LOW-POWER COMPUTATION USING FPAA FOR WEARABLE DEVICES

A Dissertation
Presented to
The Academic Faculty

by

Sahil Shah

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
May 2018

LOW-POWER COMPUTATION USING FPAA FOR WEARABLE DEVICES

Approved by:

Professor Jennifer Hasler, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Asif Islam Khan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Omer Inan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Bradley Minch
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor A. Fatih Sarioglu
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 22 March 2018

Dedicated To

My Parents

Sejal Shah

and

Sandeep Shah

ACKNOWLEDGEMENTS

I would like to thank my PhD advisor Jennifer Hasler for her advise and guidance. I would also like to thank Omer Inan for providing insight into several biomedical applications and allowing me to work with several members of his lab. Bradley Minch for the comments in my thesis defense and meticulously reading my thesis document. In addition, I am grateful to Dr. Fatih Sarioglu, and Dr. Asif Islam Khan for serving on my committee.

Dr. Jennifer Blain Christen, my MS advisor at ASU, for inspiring me to pursue research and being an excellent mentor over the years.

I would also like to thank past and present members of ICE lab. In particular, Sihwan Kim who has been there through thick and thin of the graduate life. Aishwarya Natarajan for several technical and non-technical discussions and general musing of graduate life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this research is to investigate low-power mixed-signal computation techniques for real time applications. The need for real-time processing, with rise of wearable devices, creates a strong drive for researching and developing methods and system architectures which reduces the power consumption. By performing the computation locally near the sensor node one can increase the energy efficiency of such devices by reducing the need for communication to the cloud. Analog computation has shown promising results in the space by significantly reducing the power consumption by processing the signal in analog without having to convert it into digital domain. Further, by adding programmability and configurability to analog, the effects of process, voltage and temperature variations could be reduced significantly.

CHAPTER I

LOW-POWER COMPUTATION

Since the advent of Moores Law in 1965, the number of transistors on an integrated circuit has almost doubled every year. As a result, there has been a consistent improvement in performance and speed of computation. But, with the channel lenght of a transistor almost reaching its physical limits, it has become more important to investigate and to develop new computational techniques. This has led to the development of fields like neuromorphic engineering and analog/mixed-signal computation.

Along with the above mentioned saturation of Moore's Law, there has been a significant interest in technologies like wearable devices for healthcare applications and Internet-Of-Thing (IOT) devices for remote sensory nodes. This has led to a tremendous growth in developing low-power computation techniques. Most of the effort in industry as well as academia has been to reduce the power consumption of digital processors. Their argument has typically been that digital circuits are robust and have high accuracy. But, most real-world signals are analog in nature. This fact should immediately raise questions on the precision and power consumption of the Analog-to-Digital Converters (ADC) so that the digital processors could achieve their classification/processing accuracy, and how the system would scale when there are multiple sensors/analog inputs. Also, scaling to a larger number of wearable devices, or larger amount of data generated by these devices, will increase the bandwidth required for communication, if cloud servers are solely used for computation.

These issues coupled with the need for real-time processing creates a strong drive for investigating and developing methods and system architectures that reduce power consumption and compute locally near the sensor node. Analog computation has

shown promising results in this space by significantly reducing power consumption by processing signals in analog without having to convert them into the digital domain. Further, by adding programmability and configurability to analog, the effects of process, voltage, and temperature variations can be reduced significantly. This work seeks to use a Field Programmable Analog Array (FPAA) to perform real-time processing and analog computation.

This document is organized as follows. Chapter 2 describes the basic structure and programming of Floating Gates (FG) and there use in analog signal processing. The chapter also discusses the evolution of FG Field Programmable Analog Array (FPAA). It introduces the state-of-the-art FG based FPAA which is predominantly used in this work.

Chapter 3 describes a Built-in Self Test (BIST) system to tune analog front-end used extensively for feature extraction in analog classifiers. Specifically, the capacitively coupled current conveyer circuit which perform band-pass filtering of the input signal. In general, it could be used to tune multiple parameters on a chip.

Chapter 4 describes techniques for reducing temperature variability of several different circuits on a reconfigurable platform. It also introduces models for simulating temperature behaviour of various analog circuits. Measurement from several current and voltage reference are presented in the Chapter.

Chapter 5 shows the application of reconfigurable analog platform for monitoring vital signs. Various physiological signals are analyzed using low-power analog processing techniques. In particular four different physiological signal namely electrocardiography, blood pressure, photoplethysmography and impedance pleythsmography are analyzed.

Chapter 6 introduces several different analog classifiers. It discusses a classifier which distinguishes between speech and noise signal. The chapter also shows measurement results from a system which is used as an activity detector analyzing signal

from an accelerometer. Also, a proof-of-concept classifier analyzing acoustic signals from the knee-joint to determine the presence of ACL injury.

Chapter 7 describes in detail the different aspects of implementing an analog classifier on the chip. It introduces calibration of the circuits and systems used as part of the analog classifier. The learning algorithm used for training the classifier is also described. The Chapter also shows measurement results from a VMM+WTA classification of an acoustic dataset created using a series of 1s data inputs, identifying the presence of a sound source, whether it be a generator, truck, or car.

CHAPTER II

FLOATING-GATE-BASED FPAA

2.1 Floating-Gate Transistor

The use of Floating-Gate (FG) transistors to implement non-volatile memory was first shown in 1967 by Kahng and Sze [1]. This has led to development of non-volatile memory, solid-state drives, which allow systems to store data in absence of power.

They have also been used in several analog/mixed-signal systems. The ETANN chip [2] used the FG to store weights of an artificial neural network. They were used to compute the weighted sum of inputs and operate as threshold logic gates [3]. Several other application include their use as an analog memory [4], a FG fourier processor [5], to perform offset cancelation in amplifiers [6], a variety of others.

Figure 1 shows the schematic and layout of a pFET FG transistor. The poly-silicon gate is insulated completely with SiO_2. This allows the charge to be stored on the floating-node. External inputs are coupled onto the floating-node using a capacitor (C_{in}). A tunneling junction (V_{tun}) is coupled via a tunneling capacitor (C_{tun}). To reduce the effects of trapping, tunneling is performed through a high quality oxide and hence C_{tun} is designed using the gate oxide between the gate poly-silicon and n-well.

2.1.1 Programming Of A Floating Gate Transistor

FG transistors can be programmed precisely with a combination of hot-electron injection and Fowler-Nordheim tunneling [7, 8]. In a large array this can achieved by either direct or indirect programming methods [9]. Each method has its own trade-offs and which technique should be used depends on the application. Figure 2 shows the structure of two FG programming methods. Figure 2(a) shows the programming

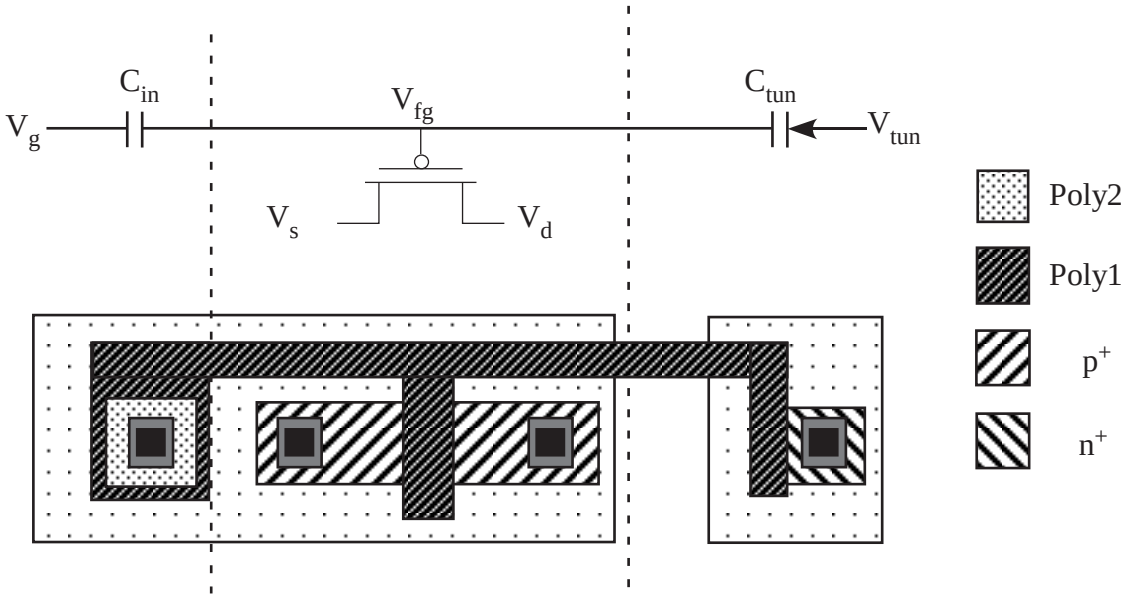


Figure 1: Circuit schematic and layout of a pFET floating-gate transistor. The floating-node(V_{fg}) is surrounded by SiO_2 as an insulator. Inputs are coupled onto the floating-node using an input capacitor (C_{in})

structure of an indirect programming method where the transistor M_p is used for active programming whereas M_a is used as part of the system or circuit. A direct programming structure is shown in the Fig. 2(b) where Transmission Gates (T-Gates) are used to switch between the two phases of operation, the programming phase and the run phase.

Which of these two methods is used depends on the application. Direct programming requires extra T-gates for programming precisely and hence requires a larger area per FG. By contrast, the indirect programming structure only requires an extra FG transistor and hence increases the density of the FG cell. Because in case of indirect programming, the programming infrastructure measures and programs the programming transistor there is a threshold voltage mismatch between the transistors and requires an additional routine for calibrating out the mismatch.

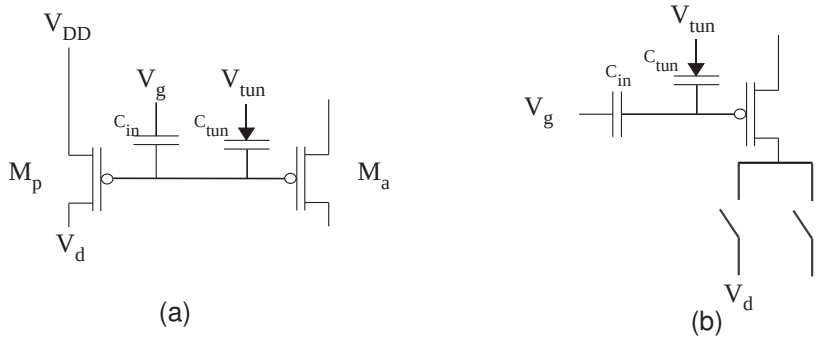


Figure 2: (a) Programming structure of a pMOS FG indirectly programming another pMOS FG. Transistor M_p is connected to the integrated programming infrastructure. The infrastructure measures and programs the transistor M_p using hot-electron injection. (b) Programming structure of pMOS FG and T-gates for directly programming the FG. T-gates are used to switch between the programming phase and the active phase.

2.1.2 FG programming using integrated programming infrastructure

Figure 3 shows the algorithm steps for programming a FG transistor. The infrastructure uses Fowler-Nordheim tunneling as a global erase before starting a new sequence of precise programming. The block diagram at the top of Fig. 3 illustrates the steps in the process from loading the programming code onto the SRAM to using the programmed system in an application.

We initially tunnel all the FG transistors to ensure that a FG pFET device has no channel current. This erases the previous design compiled on the FPAA. Then we perform a reverse-tunneling operation, to increase the charge on the FG node, to the point that the pFET has a small, but negligible, channel current.

After reverse tunneling FG pFET devices that we do not program will stay in accumulation and pass negligible levels (e.g., $\leq$1 pA) of current, even for scaled-down devices. The next step is to perform recover injection. Here, the FG devices are injected while measuring their drain current. Depending on the target current FG devices are injected to either have 1nA (for a target current between 30pA to 1nA) or 20nA (for target current above 1nA) of drain current while biasing the gate (V_g)

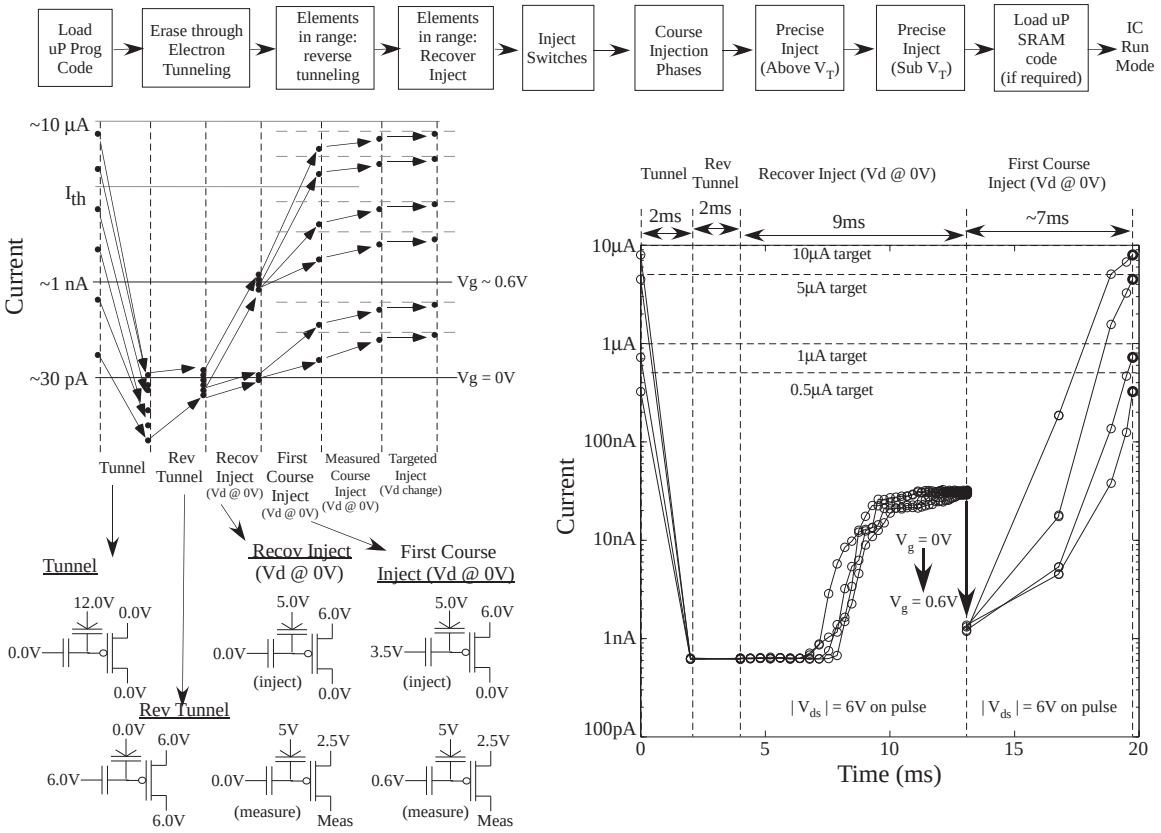


Figure 3: Algorithm steps for programming an array of FG devices. Top: flow chart of the key required programming steps. Left: graphical illustration of the FG programming steps following the procedure for a range of FG devices starting and ending at a desired target value. Right: experimental data showing actual trajectories of four FG devices through the first four programming steps. The goal is to get devices close to target current. During run-mode the FG devices have a V_g 0.6 V. For target current below 1nA the measurement of FG devices is performed using V_g of 0 V.

with 0 V. While operating the FG transistors they are biased at a gate voltage of 0.6 V.

Target programming typically requires measuring channel current, comparing it with the desired current, performing a range of calculations for the different parameters (e.g. drain voltage) during the next injection pulse, and repeating the process until it sufficiently converges. The method of calculating these parameters is described with detail in [10]. Coarse programming steps use these parameters to be in the vicinity of the target value. Figure 3 shows two sets of coarse injection, first

coarse and measured coarse injection. First coarse injection involves injecting with calculated parameters without measuring. The next step involves coarse injection with measurement of drain current and hence it is called measured coarse. Both of these coarse injection use the same parameters but the difference is that during measured coarse the channel current is closer to the target value and hence repeated measurements are performed so as not to overshoot the target current.

The final step involves precise targeted FG injection. The approach starts by measuring the channel current and comparing the result with the target value. Based on this, parameters such as drain voltage and gate voltages are calculated to reach the desired channel current. Then the system applies injection pulse with calculated parameters and the new channel current is measured. This process of injecting and measuring is repeated till sufficient accuracy is achieved. This algorithm enables achieving a precision of almost $1 - 2\%$ [10].

2.2 Field-Programmable Analog Array

Field-Programmable Analog Arrays (FPAAs) are poised to revolutionize analog and neuromorphic systems the same way that FPGAs revolutionized digital systems, by making prototyping cost effective and shortening the test cycles [14, 15]. Also, FG based FPAAs [16], due to their reconfigurability, have the potential of operating beyond the energy efficiency wall [17].

The first FPAA was built in 1991 [26]. A FG based analog array was introduced in [18]. A digitally enhanced reconfigurable platform was shown in [12]. The first mixed-signal FG based reconfigurable array was shown in [13]. The current FPAA integrates concepts from these several different FPAA architectures, as shown in 4. Table 1 shows the evolution of FG based FPAAs:

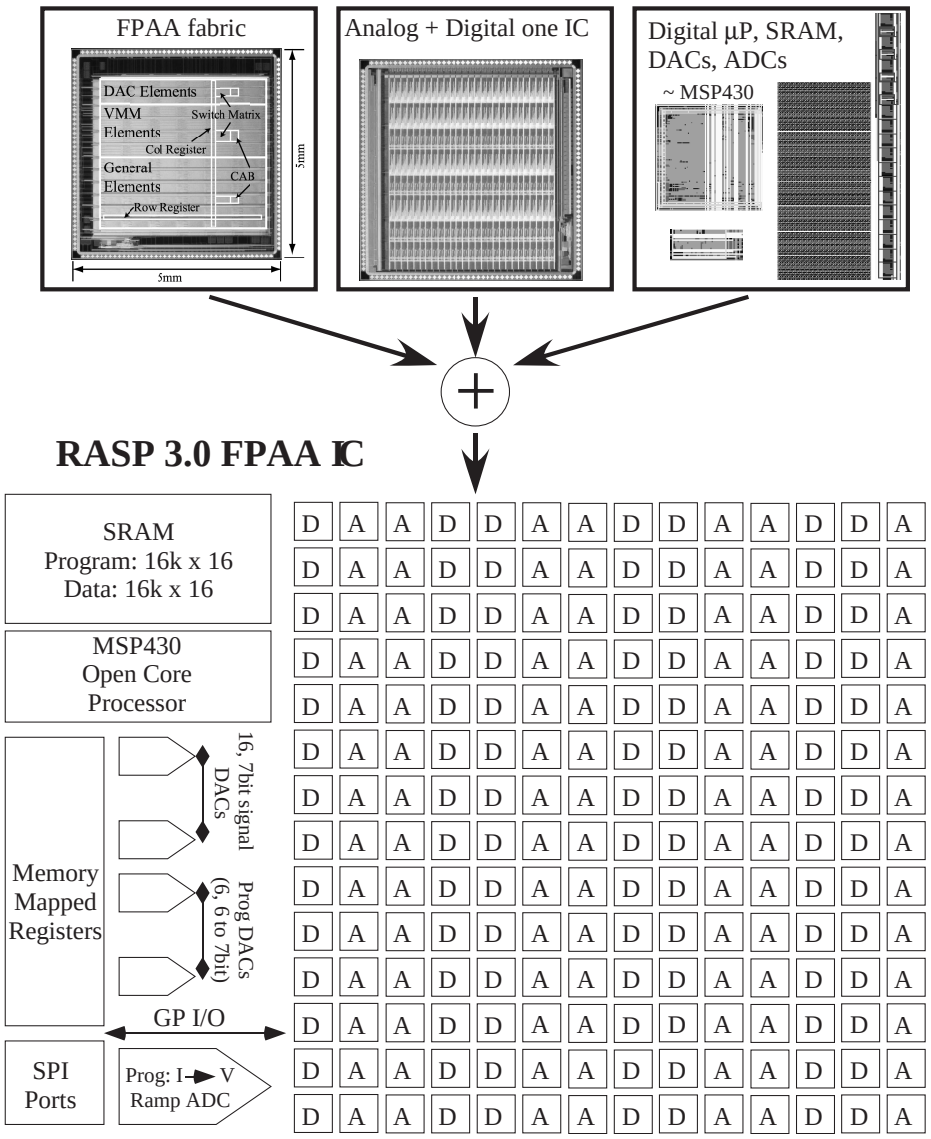


Figure 4: RASP 3.0 integrates divergent concepts from previous multiple FPAA designs [11–13].

2.2.1 ARCHITECTURE DESCRIPTION OF THE FPAA SoC IC

Our mixed-signal FPAA is shown in Fig. 5, having both analog elements, Computational Analog Blocks (CABs), and digital elements, Computational Logic Blocks (CLB). The FPAA consists of 98 CABs and 98 CLBs. CABs and CLBs are connected using manhattan-style routing composed of Connection (C) and Switch (S) blocks. These interconnects allow analog and digital blocks to interact with each other thus

Table 1: Evolution of FG-based FPAAs

Year	Reference	Discussion
2002	[18]	Floating-gate Switches for routing
2005	[19]	RASP1.5 with 2 CABs and FG crossbar
2006	[20]	RASP2.5 with 56 CABs
2009	[21]	RASP2.7
2010	[22]	RASP2.8 with 32 CABs
2012	[23]	Floating-gates as VMM
2012	[24]	RASP2.9v with 78 CABs
2013	[13]	First mixed signal FPAA
2016	[25]	RASP3.0 with 98 CABs and CLBs

leveraging computational capabilities of both analog and digital circuits. Interconnect switches are composed of nonvolatile FG transistors that are programmed using hot-electron injection and globally erased using Fowler-Nordheim tunneling. The programming infrastructure, composed of DACs and an ADC, is controlled using a low-power, open-source MSP430 microprocessor. The microprocessor has a controllable frequency of 0-50 MHz. An $8K \times 16$ SRAM is used to store the program to be executed by the microprocessor and a separate data memory of $8K \times 16$ is also present. As a part of the infrastructure, there are sixteen 7-bit DACs for generating signals. These DACs can be routed to the FPAA fabric to work as an arbitrary waveform generator. The 14-bit ramp ADC, which is used for measuring the current of a floating gate, during the programming phase of the FPAA, can be routed to the FPAA fabric during run mode. In that case, it would behave as a data acquisition device and store its outputs on the available data memory.

Figure 5 shows basic elements of a CAB. Inputs and outputs of the CAB elements can be connected via a local routing, instead of a global routing, which has a reduced parasitic capacitance associated with it. A CAB consists of multiple Operational Transconductance Amplifiers (OTA) with the ability to select between wide linear range and high gain amplifier. Current bias of the OTA is set using a FG pFET

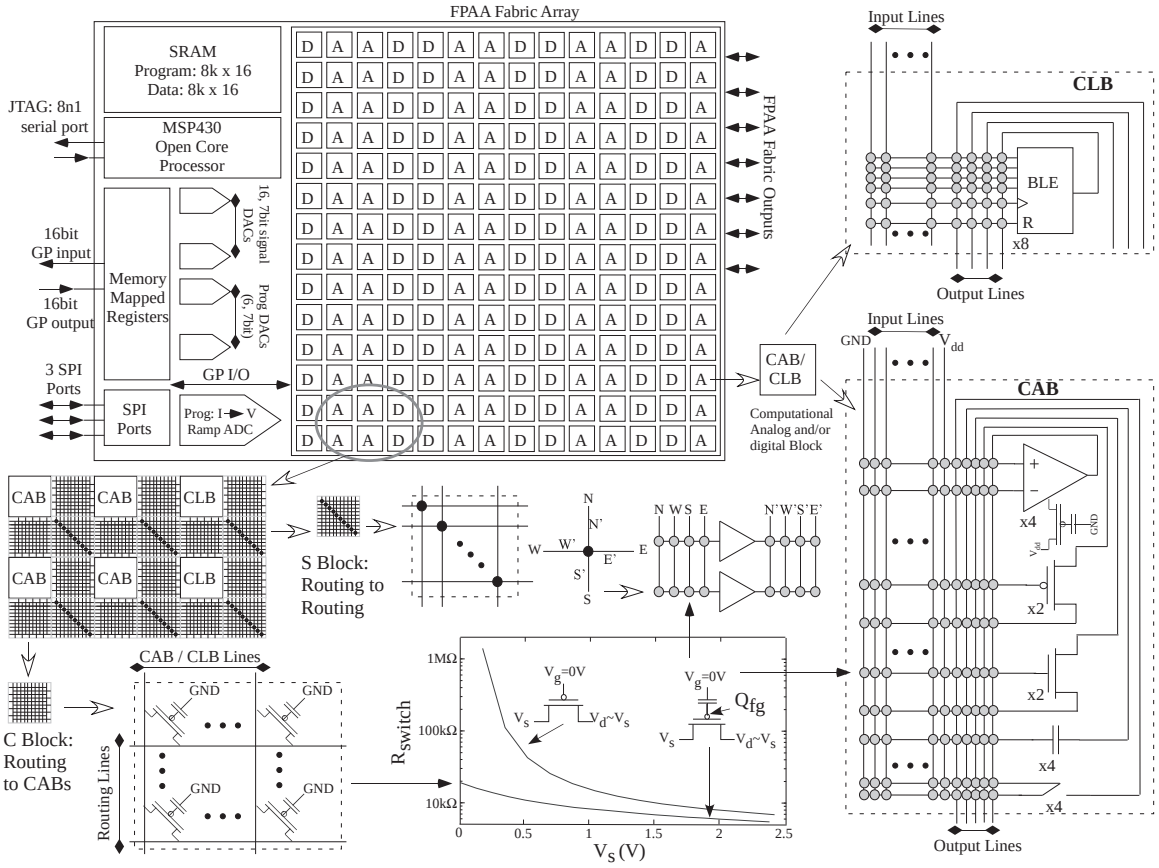


Figure 5: RASP 3.0 functional block diagram illustrating the computational blocks and routing architecture. The infrastructure control includes a microprocessor developed from an open-source MSP430, as well as on-chip structures including the on-chip DACs, current-to-voltage conversion, and voltage measurement, to program each FG device. The FG switches in the connection (C) blocks, the switch (S) blocks, and the local routing are a single pFET FG transistor programmed to be a closed switch over the entire fabric signal swing of 02.5 V. The CABs and the CLBs are similar to previous approaches [13,27,28]. Eight, four input BLE lookup tables with a latch comprise the CLB blocks. Transconductance amplifiers, transistors, capacitors, switches, and other elements comprise the CAB blocks.

transistor. The programming infrastructure enables precise programming the bias current values from 30pA to 10μA. Thus, an FPAA allows for multiple parameters, like linearity, gain, and power, to be tuned, depending on the application.

2.2.2 The Mixed-Signal Aspect

The FPAA fabric enables seamless integration of CABs and CLBs. This makes it attractive for mixed-signal applications and takes advantage of both computational domains. Certain applications require processing in the digital domain and one could use verilog code to compile logic elements on the SoC. This has been made possible due to the development of both high-level [29] and low-level tools [30].

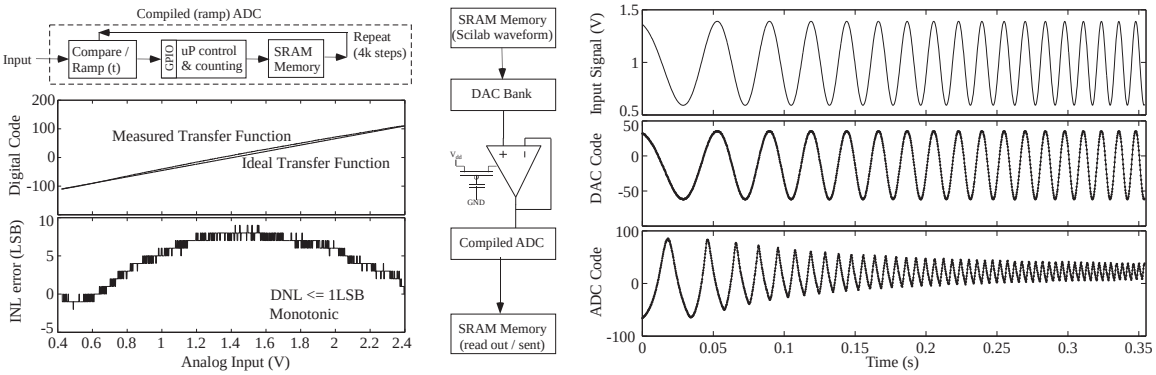


Figure 6: Measurement of a compiled 8-bit ramp ADC used in the FPAA device. 4kSPS ADC for this 8-bit ramp conversion integrates the compare + ramp functions compiled into a single CAB, routing the digital output bit to a GPIO. This simple ADC polls for the digital bit to change while counting in the processor. A compiled system for a first-order low-pass filter using these components, where the output load capacitance is implicit (output going into the ADC), because of the node capacitance from the routing infrastructure. The programmed corner frequency of the LPF is 150Hz. The device is measured by applying a linear chirp input signal going from 25Hz to 250Hz, showing the signal attenuation, as expected, as well as the signals from the input 7-bit DAC as well as the 8-bit ramp-ADC (4kSPS). The output of the ADC inverts the resulting response from the original signal.

A classic example of a mixed-signal circuit is an ADC. Figure 6 show a single-slope ramp ADC compiled on the FPAA. Figure 6 shows the characterization of the ramp ADC. The sampling rate of the ADC is 4KSPS and was designed to have a 8-bit output. Owing to the reconfigurability of the system, the ADC could be configured to have a higher sampling rate at the expense of the accuracy. The power consumption of the ramp ADC is $0.77\mu W$. Figure 6 also shows the characterization of the DAC-LPF-ADC system. The corner frequency of the first-order LPF is 150 Hz. The device

is measured by applying a linear chirp input signal going from 25Hz to 250Hz, showing the signal attenuation, as expected, as well as the signals from the input 7-bit DAC as well as the 8-bit ramp-ADC (4kSPS). The output of the ADC inverts the response from the original signal.

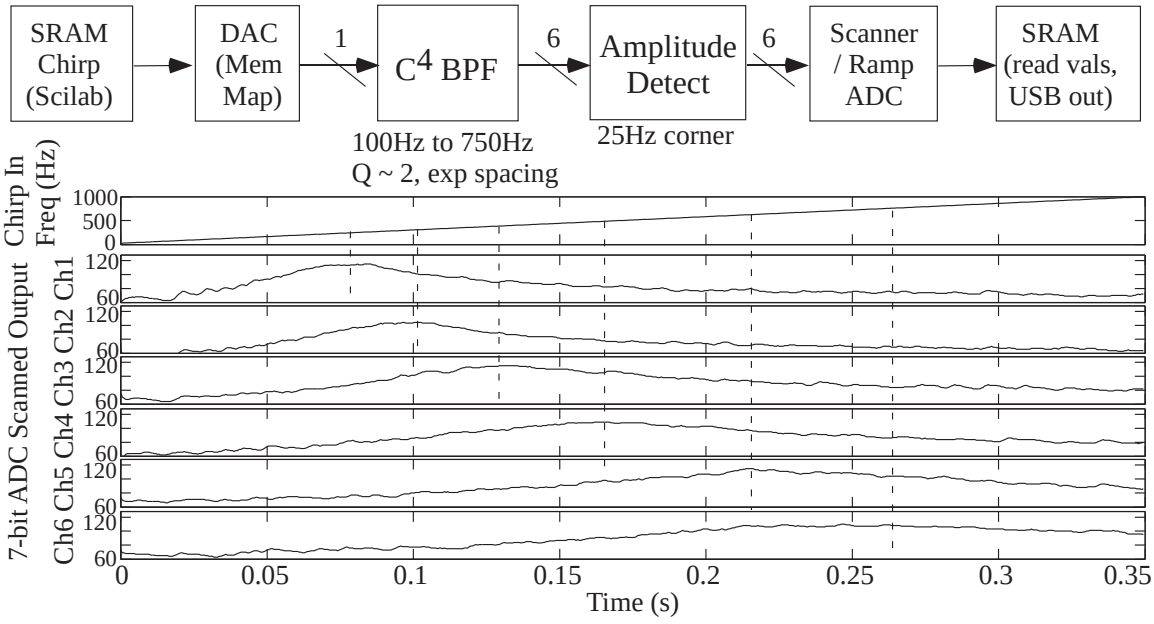


Figure 7: Block diagram (similar to Scilab / Xcos definition) and measured outputs from a bank of 6 Band-Pass Filter (BPF) and Amplitude Detect elements compiled onto and measured through a remote FPAA board. The input chirp signal, a linear sweep between 25Hz and 1kHz, is stored in on-chip SRAM to be played through a memory-mapped DAC. The output signals went through a demultiplexing module, compiled using shift-registers located in the routing fabric, and then through the same 8-bit ramp ADC module; this measurement only used the upper half of the positive values (effectively 6bit). The linear frequency sweep with time is illustrated, as well as the resulting outputs of all 6 frequency channels, each programmed to a different frequency location (exponentially spaced).

Figure 7 shows a more complex example using the FPAA system, a parallel bank of bandpass filters and amplitude detectors typically used for low-power sub-band analysis, as another example of co-design between assembly, analog, and digital components and interfacing between them. The input vector is stored in the SRAM, which is used by the DAC to output a chirp waveform to the system of 6 Band-Pass Filters. The center frequencies are programmed between 100Hz and 750Hz, seen by

the different peaks in the chirp response in Fig. 7. The output is then measured serially for each band-pass filter using shift register blocks which are part of the local interconnect. A shift register allows measuring of 16 different inputs/outputs serially and is controlled by the microprocessor. Measurements of the ramp ADC in conjunction with the shift register are shown in Figure 7.

CHAPTER III

BUILT-IN SELF TEST SYSTEM FOR LARGE-SCALE MIXED-SIGNAL SOCS

3.1 Built-in self test on FPAA

Large-scale analog, mixed-signal and neuromorphic systems suffer from mismatch and devices variations. Generally, such mismatch and variations are mitigated using large devices or complex layout techniques. In the case of FG based FPAAs there are several thousand parameters available to the user, which could be a boon if done correctly or could turn out to be an ordeal if not done right. Hence, there is a need to develop a system that can tune and adapt these parameters based on system specifications.

This chapter develops a Built-in Self Test (BIST) system to tune an analog front-end used extensively for feature extraction, the Capacitively Coupled Current Conveyer (C4) circuit which performs band-pass filtering of an input signal. The use of second-order-section, a filter with several parameters, in emulating a silicon cochlea has evolved over the years from the first analog silicon cochlea [31], to an improved version where the silicon cochlea has increased linearity and stability [32]. Moreover such filters have also been used in active 2-D cochlea with nonlinear properties of biological cochlea [33], to their use in more recent large scale binaural spatial audition sensor [34]. The C4's have also seen their use as a Fourier processor, and as a front end for speech detection [35,36]. Such large scale neuromorphic system would require the tuning of a few hundred parameters precisely. In the case of reconfigurable analog systems, such as the one used for speech processing [37], the number of parameters that need to be tuned are more than 50,000 [25]. The tuning algorithm proposed in this chapter could be generalised for such large systems and could lead to decreased

testing time and efficient computing.

One of the major issues in implementing an on-chip continuous-time filter is to keep its frequency response stable. It has been shown that due to mismatch and variation in the fabrication process the frequency response could vary by up to 50% [38]. In the case of multiple filter banks or a larger system these variations and mismatch often lead to lower efficiency and higher design margins, thus the need for an automatic Built-In Self Test (BIST) system which could reduce the variation and mismatch while implementing multiple filter banks. Figure 8(a) shows the usual approach for such an automatic tuning system. This approach involves tuning a single band-pass filter and reducing either variation in center frequency or quality factor.

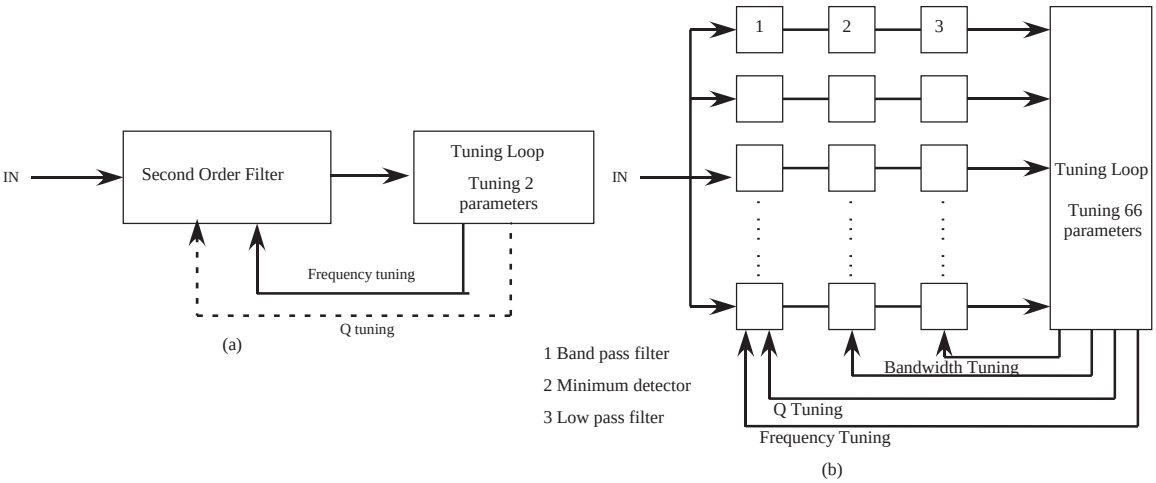


Figure 8: System diagrams for a BIST. (a) Typical implementation of a BIST system for tuning quality factor and center frequency. This approach involves tuning a single parameter for a single band-pass filter using an off-chip reference. (b) The proposed system for BIST to tune multiple parameters. The system is implemented on a mixed-signal FPAA. The proposed algorithm tunes 66 parameters. These are the center frequency, quality factor, gain at the center frequency of the Band-pass Filter, DC offset and bandwidth of amplitude detectors, and the time constant of LPFs.

This chapter describes a system that can automatically tune multiple filter banks. The systems reduces the variation/mismatch in center frequency, quality factor, gain at the center frequency of the band-pass filter. It also reduces the variation in bandwidth of amplitude detectors and time constant of Low Pass Filters (LPF). The DC

offset of the filter bank chain is also tuned. In general, it could be used to tune multiple parameters on a chip. Figure 8(b) shows such a system where these parameters are tuned using a tuning loop. Here, a set of 12 parallel second-order band-pass filters [39] are used as Devices Under Test (DUT). This system is compiled, using open-source Xcos/Scilab tools [29] onto a large-scale FPAA [25] and routed using a modified version of Versatile Place and Route (VPR) [40]. Figure 8(b) also shows amplitude detectors and LPFs as a part of the filter bank chain. Parameters such as the bandwidth of amplitude detectors and time constant of the LPFs are also tuned. The system is then tested on three different FPAA chips to evaluate the accuracy of the algorithm.

In Section 3.2, the variation and mismatch found in a FG-based FPAA is described, in the context of designing a larger system is discussed in detail. The proposed algorithm and its tuning capabilities are described in Section 3.3. The use of this algorithm to tune filters on three different FPAAs is discusses in Section 3.4. The deviation of each parameter from its mean is also presented in this section. Section 3.5 summarizes the performance of our system and compares it to other adaptive continuous-time filters. This section also discusses the use of such a system in large-scale neuromorphic and biomedical systems where power consumption and efficiency are important factors.

3.2 Mismatch, Variation and Programming the G_m

As the feature size of a CMOS process scales, mismatch in the threshold voltages (V_{T0}) of transistors and parasitic capacitances can result in significant overhead in designing a larger system. One of the hypothesized factors for the energy efficiency wall is due to larger components used for reducing the variation [17]. Energy Efficiency wall is the saturation of energy efficiency resulting from scaling of digital processors. In the case of an FPAA or an FPGAs the variations also depend on placement and routing

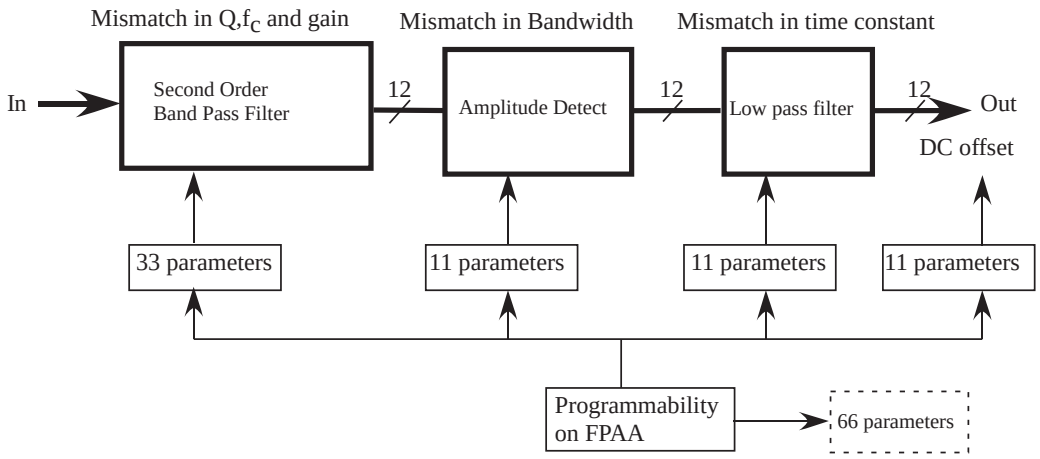


Figure 9: Mismatch in different parameters of a filter bank chain. The output of the blocks are vectorized, with N=12. An example of variation in two parameters, in this case f_{-3dB} of the LPFs, is shown. For the continuous-time filter the number of parameters to be tuned are 66. In general, the number of these parameters could be more than 1000, especially in a reconfigurable system where available parameters are 50,000, which really requires an automated tuning system.

of components. The problem of variation and mismatch is usually addressed by using larger transistor sizes, using common-centroid methods to layout critical components and various DC offset cancellation techniques [41]. Variation and mismatch in the transistor can be easily mitigated by injecting a charge on to the floating gate [10]. Figure 9 shows some of the variations and mismatch found in our system. The parameters shown here are a subset of mismatch and variation found in a large-scale reconfigurable system. In general, the number of parameters could be larger than 1000 in an ASIC and about 50,000 in a reconfigurable FPAA. Thus for an efficient use of resources, an automatic tuning and a calibration system that can tune multiple parameters is necessary.

In the case of a continuous-time filter implemented on an FPAA, the source of mismatch is due to variations in parasitic capacitance, which is a part of the local routing in the CAB, depicted by C_L in Fig. 10(a), variations in global routing based on the placement by VPR shown in Fig. 11, and mismatch between two transistors used for indirect programming of the floating gate shown in Fig. 10(b). Indirect

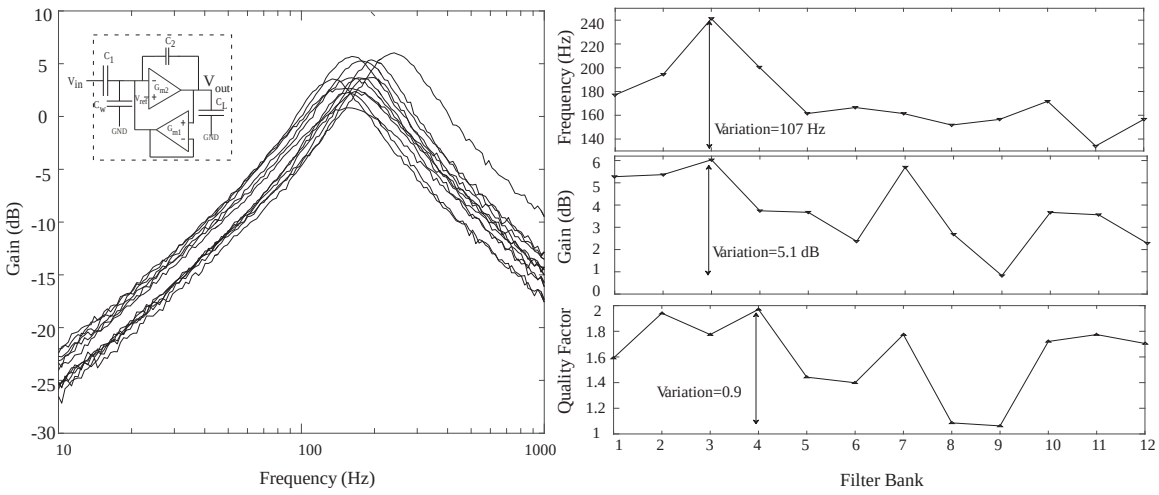


Figure 10: Typical variation and mismatch found in an FPAA. The frequency response of 12 filter banks programmed with the same bias current.Also, the variation in quality factor, center frequency, and gain at the center frequency is shown. These variation are due to mismatch in the current biases of the OTA, local mismatch between the programming transistor and the one used in the circuit, and mismatch in the capacitance.

programming of the transistor [9] reduces the parasitic capacitance and extra switches for programming at the cost of increased threshold-voltage mismatch. The transistors used for biasing the OTA are relatively small (W/L=$\frac{6\mu m}{2\mu m}$), to increase the density of the CAB, and no special layout techniques have been used for reducing the mismatch, because the programmability of the floating gate can compensate for this mismatch. The circuit used for the second-order band-pass filter is shown in Fig. 10(a). The transfer function of the band-pass filter is given by (1)

$$\frac{Vout}{Vin} = \frac{s^2 C_1 C_2 - s G_{m2} C_1}{s^2(C_p C_T - C_2^2) + s(G_{m1} C_p + G_{m2} C_2 - G_{m1} C_2) + (G_{m1} G_{m2})} \quad (1)$$

In (1), C_p is a summation of C_L, load parasitic capacitance, and C_2. The feedback capacitance C_2 could be a parasitic routing capacitance to reduce the biasing current and thus the power consumed but in this work we have used an explicit capacitor from the CAB. The input and output of a CAB having some parasitic coupling capacitor resulting from the CAB and local interconnect structure. C_T is the summation of

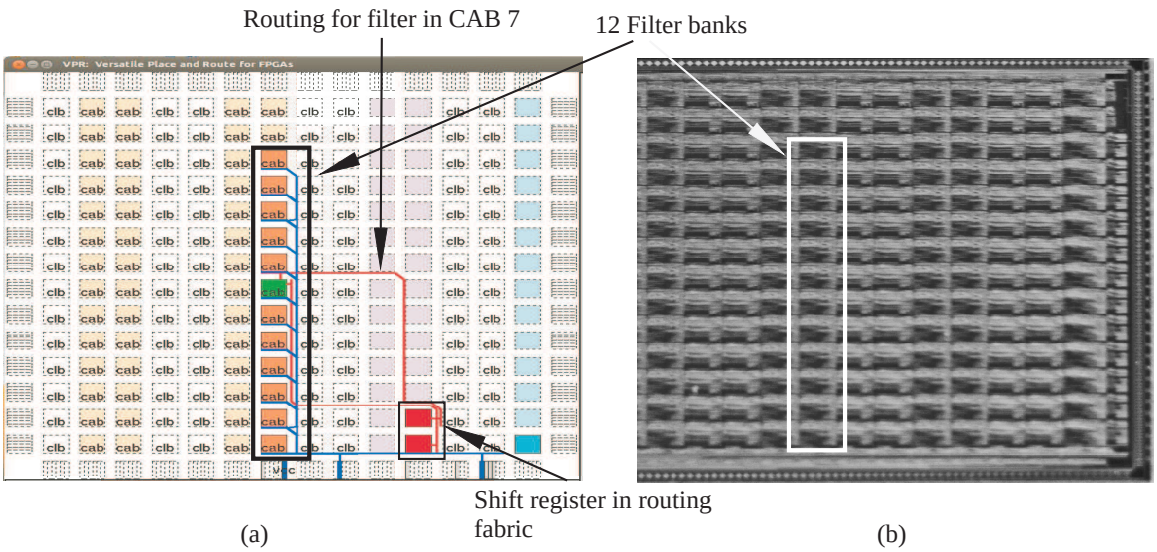


Figure 11: (a) Output of a modified versatile place and route [40]. Placement of 12 parallel filter banks and their routing to 16-bit shift registers implemented in the routing fabric of the FPAA. There are two shift registers used here for characterizing the frequency response of the filters after tuning. Load capacitance for each filter bank, due to the routing length and the number of C and S blocks used, is different for different filters and is one of the sources for variation. (b) Die photograph of a SoC FPAA. Twelve CAB used are highlighted on both VPR output and the die photo.

C_2, C_W (parasitic capacitance at the input of G_{m2}), and C_1 which is the input capacitance. The transconductance G_{m1} sets the low frequency pole and feed-forward transconductance G_{m2} sets the high frequency pole. The quality factor and gain at the center frequency of the band-pass filter are respectively given by

$$Q = \frac{\sqrt{C_T C_p - C_2^2}}{C_L\sqrt{\frac{G_{m1}}{G_{m2}}} + C_2\sqrt{\frac{G_{m2}}{G_{m1}}}}$$

$$A = \frac{-C_1}{C_2}\frac{1}{1+\frac{G_{m1}C_L}{G_{m2}C_2}}$$

The time constant for the low frequency pole and high frequency pole are respectfully given by

$$F_{low} = \frac{C_2}{G_{m1}}$$

$$F_{high} = \frac{C_T C_p - C_2^2}{G_{m2} C_2}$$

Because each of these parameters depend on the G_ms of the OTAs, we can compensate the Q, gain, higher and lower frequency pole of the band-pass filter by programming the transconductance in the right way.

The circuit is built using standard components present in a CAB described in Chapter 2 and is fully reconfigurable, as opposed to a custom design. The transconductance used in the filter is the 9-T OTA structure shown in Fig. 10(b) and has a floating gate input to compensate for any input DC offset and also allows for a wider linear range. Thus, for a given bias current transconductances G_{m1} and G_{m2} are smaller compared to that of a non-Floating-Gate OTA (FGOTA) because of the presence of a capacitive divider at the input of an FGOTA. Here, a capacitor of 192fF is used at the input of the FGOTAs.

Figure 10(c) shows variation in the frequency response of 12 parallel continuous-time filters, placed by VPR tool, as shown in Fig. 11(a). Continuous-time filters in Fig. 10(c) were programmed using the same current values, by measuring indirectly the bias current of the programming transistor (i.e, not the transistor in circuit). FG transistors are calibrated for global variation [42]. Calibration allows for compensation of global mismatch in the FPAA fabric, as well as mismatch in the programming infrastructure. Thus the variation, seen in Fig. 10(c), is primarily because of the mismatch in parasitic capacitance, the local threshold-voltage mismatch between the programming and the transistor used in the circuit, and (to a certain extent) the finite resolution of the measurement infrastructure. Figure 10(b) shows the two pFET structure used for indirect programming, which is the source of local threshold-voltage mismatch. Figure 10(d) shows the variation in quality factor, center frequency, and gain at the center frequency. The variation values reported here are the difference

between the maximum and minimum values. A variation of 107 Hz in center frequency, 5.1 dB in gain at the center frequency, and 0.9 in quality factor of the filters was observed. These variations could lead to significant errors while using them for analog computation. Also, compensating for these mismatches and variation without an automated system would be tedious, time consuming and error prone. As systems scale to a larger designs, the number of tuning parameters would increase, and thus there is the need for an automated tuning system that can handle multiple parameters over multiple chips.

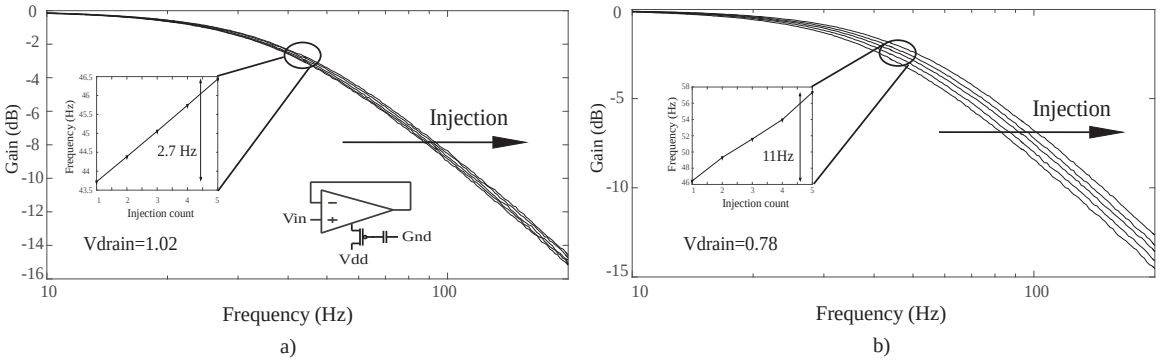


Figure 12: Measured changes in the frequency response of a LPF with hot-electron injection. The inset shows the configuration in which the OTA was used for the experiment. During hot-electron injection the source voltage of the FG transistor is held at 6 volts. (a) Variation of frequency with injection for a source-to-drain voltage of 4.98 V. This allows us for a finer control over the change in frequency of the pole. The change in f-3dB frequency with injection is also plotted in the inset (b) Variation of frequency with injection for a source to drain voltage of 5.22 V. This allows us for a coarser control over frequency of the pole.

To demonstrate the effect of injection on frequency, a LPF composed of an OTA in a follower configuration, as shown as an inset in Fig. 12, was used. Figure 12 shows the change in frequency response with injection for a set drain-to-source voltage of the biasing transistor. The subthreshold current through a floating gate transistor, used as the biasing transistor of an OTA, in the saturation regime is given by (2) [43]

$$I_s = I_{th} e^{\left(\kappa_p (V_{dd} - V_{fg} - V_{T0})\right)/U_T} e^{(V_{dd} - V_s)/U_T} \quad (2)$$

where κ_p is the fractional change in pFET's surface potential due to the change in V_{fg}, and U_T is the thermal voltage. V_s is the source voltage of a floating gate pFET. The source voltage of a floating gate transistor is set to 6 V during injection. Depending on the distance from the target frequency, either a drain voltage of 1.02 V or 0.78 V is used. The drain voltage, for all the programming floating gate transistors on the FPAA, is set using a DAC controlled by the processor. Higher drain voltage, thus a lower source-to-drain voltage, allows for a finer control over the frequency, where as a lower drain voltage will allow us to reach the target faster. A simple model for hot-electron injection oxide current is given by $I_{inj0}\left(\frac{I_s}{I_{s0}}\right)^{\alpha} e^{-\Delta V_d/V_{inj}}$ where I_{inj0} is the injection current when a floating gate operates with current reference I_{s0}, V_{inj} is a device and bias dependent parameter, and α is $1-\frac{U_T}{V_{inj}}$. The inset in Fig. 12(a) and (b) show a change in frequency at -3dB attenuation of the LPF with each injection pulse. A Vdrain of 1.02 V resulted in a change of 2.7 Hz, over 5 injection pulses of 20μs duration. This allows for a finer control over the frequency. A Vdrain of 0.78 V results in a change of 11 Hz, which is used for a coarser control over the frequency. In the case of Fig. 12, tuning was performed in open loop to demonstrate the effects of injection on the bias and frequency. In the proposed algorithm, tuning is performed in a closed loop, where after each injection the distance from the target is calculated by measuring the amplitude at the target frequency.

3.3 Algorithm for mismatch compensation

The tuning algorithm measures the amplitude at the output of the filter bank chain to determine the distance from its target frequency. The core of the algorithm is shown in Fig. 14(a). Since an amplitude detector and an LPF are used for measuring the amplitude, they have to be calibrated and tuned before tuning the band-pass filter. Reconfigurability of the FPAA allows us to test them separately and store the tuned parameters in SRAM. Hence, as seen in Fig. 14(a), after calibrating the minimum

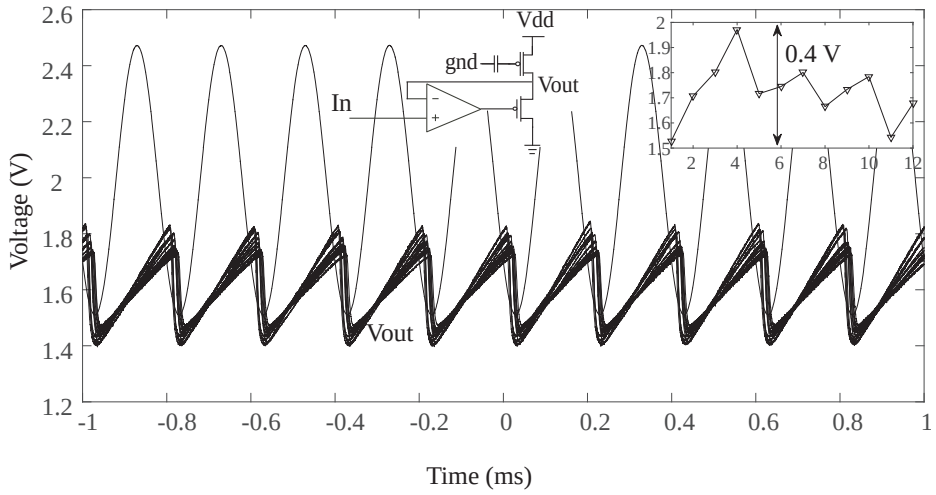


Figure 13: The outputs of 12 parallel minimum detectors after tuning their bandwidth are plotted along with its input. The dc variation is subtracted from the transient response and plotted in the inset. A maximum variation of 0.4 V was observed, which is tuned when the system is compiled with the band-pass filter and LPF.

detector and LPF, the compiled designs are tunneled, that is, a global erase is done on the FPAA fabric.

The bandwidth of the minimum detector should be above the passband of the band-pass filter. Here, all of them are tuned to operate at a maximum input frequency of 5kHz. Figure 13 shows the output of 12 parallel amplitude detectors tuned to work at 5kHz. A maximum variation of 0.4 V was observed in the output DC value before tuning. This DC variation is tuned when the whole system is compiled. If the center frequency of each filter is known a priori, the amplitude detectors could be tuned accordingly to save power. That is each amplitude detector could be tuned individually to operate just a little above the center frequency of the band-pass filter.

The next step involves compiling the tuned amplitude detectors with the LPFs and then tuning their time constants. We are interested in the time constant of the LPF rather than its bandwidth because in this configuration it is used to reduce the ripples at the output of the filter bank chain. Figure 14b shows the schematic of a minimum detector and a LPF. Initially all the LPFs are biased at a very low

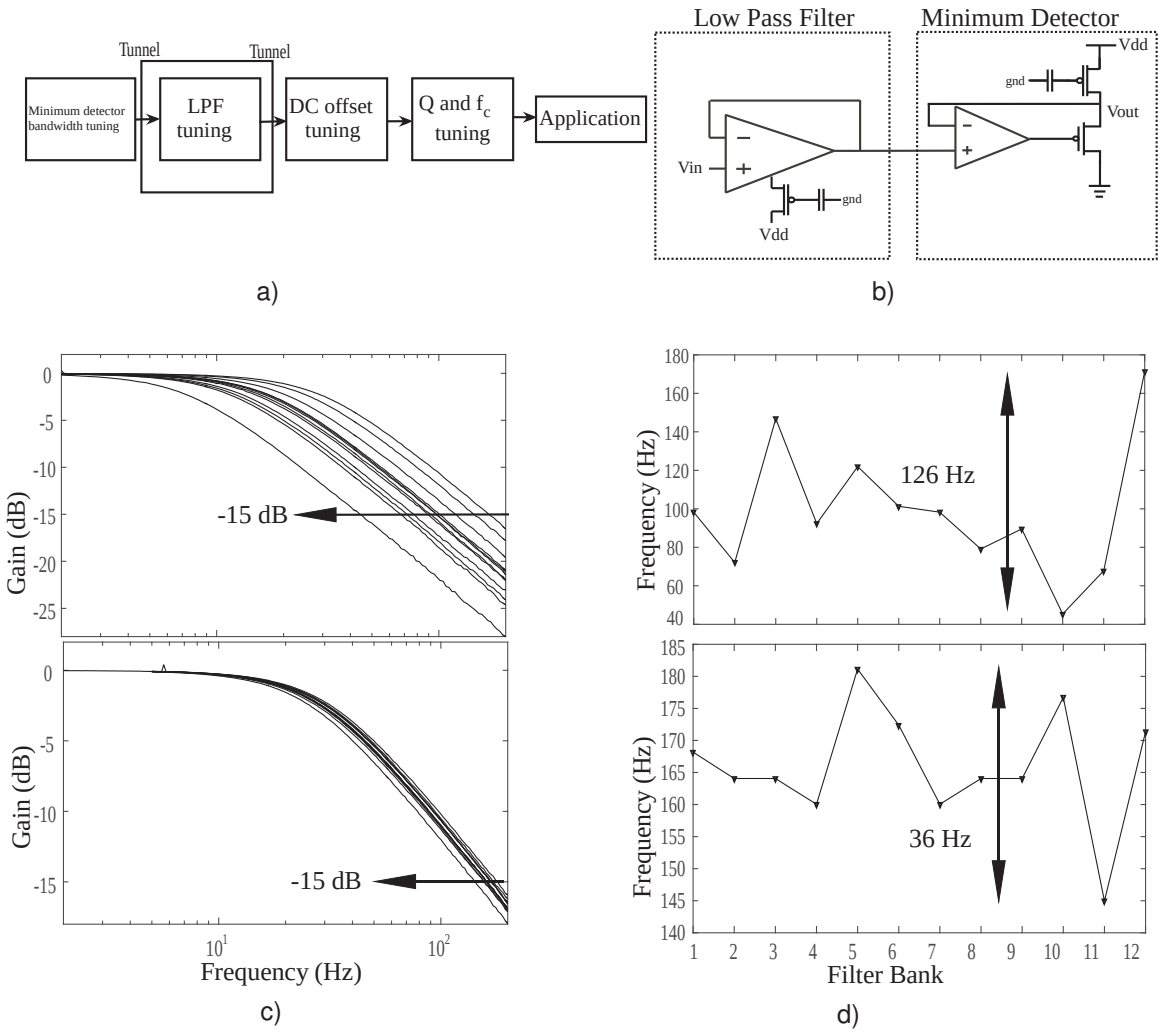


Figure 14: (a) Algorithm used for tuning the filter banks; in particular, results for LPF tuning are shown here. (b) The circuit schematic for the LPF and the minimum detector. (c) Frequency response of LPF banks. The LPFs are characterized with a bias current of 0.6nA. Tuned responses are shown below where the variation is low. (d) Variation in frequency of LPFs, at -15 dB attenuation, before and after the tuning algorithm was applied. The time constant is calculated using the frequency at -15 dB attenuation.

frequency except for the first filter, which is used as a reference and biased with the target time constant. The response of the LPF follows (3)

$$|LPF| \;=\; \frac{1}{1+\tau s} \approx \frac{1}{\tau s} \tag{3}$$

It is easier to consider the amplitude at -15 dB attenuation since the time constant (τ) of the filter is of the interest. The output of the first filter is measured at -15

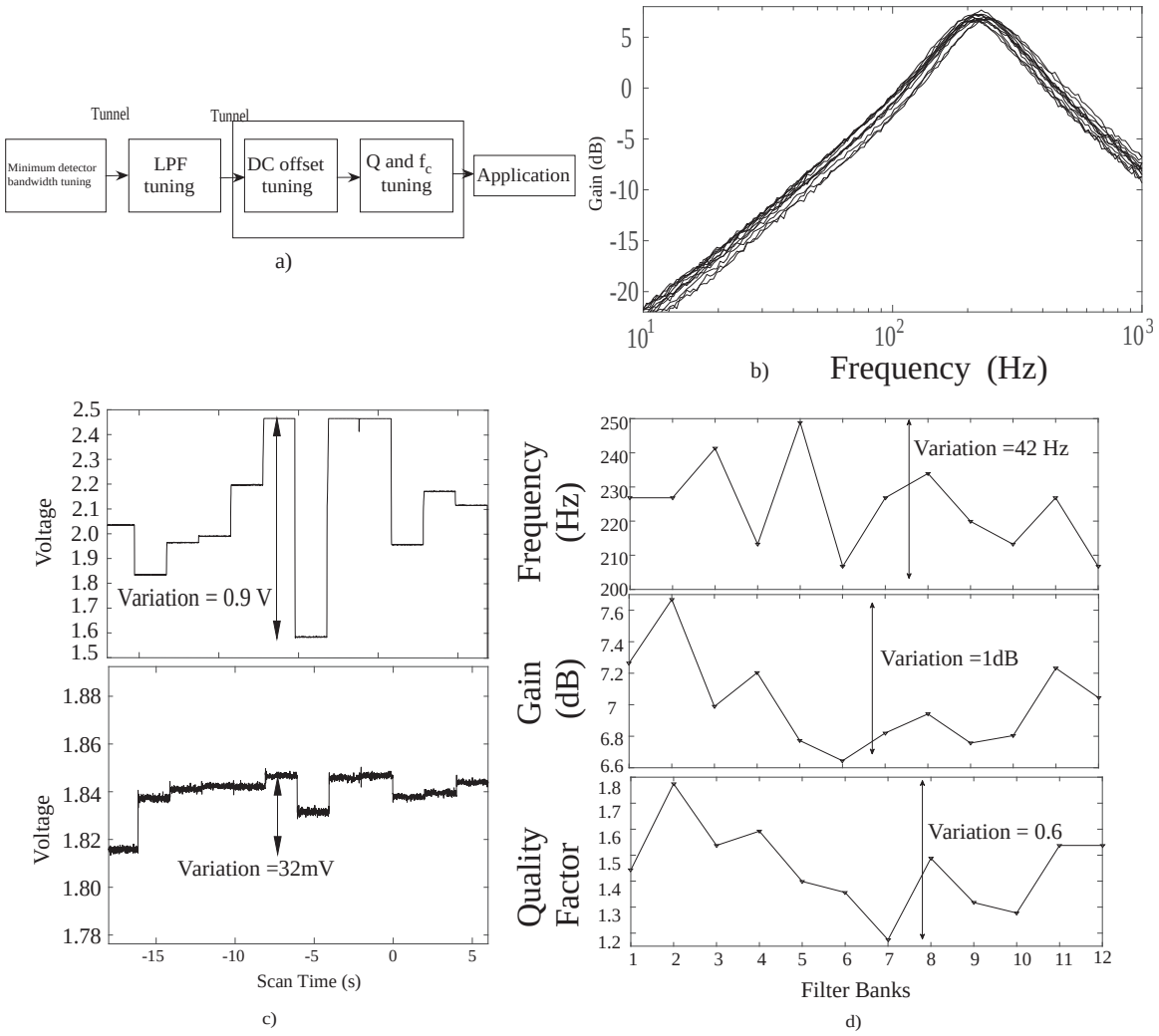


Figure 15: (a) Algorithm used for tuning the filter banks; in this case, results from tuning of DC offset, Q, and f_c are shown. (b) Frequency response of the tuned band-pass filter is plotted. (c) DC offset of the signal chain is characterized here. DC offset is compensated by tuning the FGOTA, having transconductance of G_{m2}, used in the band-pass filter. After compensation the variation in DC offset is reduced to 32 mV. (d) Variation in quality factor, center frequency and gain at the center frequency after tuning. Quality factor varies by 0.6 as opposed to 0.9 without tuning. Center frequency variation is reduced to 42 Hz compared to 106 Hz and variation in gain at the center frequency is reduced to 1 dB.

dB of attenuation, using a 14-bit ADC, and stored in SRAM. This value is used as a reference for tuning the other LPFs. Figure 14(c) shows the frequency response of the LPFs. The plot on top is of LPFs biased with the same current value, to show the variation in frequency response without tuning. The bottom plot shows of the results

obtained after using the proposed tuning algorithm. The tuning algorithm starts at a lower frequency and programs the biases till the target frequency is achieved, within certain error margin. After each injection, the output amplitude is measured and the distance from the reference filter is calculated. Based on this distance, a Vdrain value is selected for the injection, to reach the target faster or to have a finer control over the frequency/time constant. Fig. 14(d) shows the variation in LPF response, which is reduced from 126 Hz to 36 Hz at -15 dB attenuation. Generally the time constant (τ) of the reference filter could be selected according to the application. For application where the frequency spectrum of input signal is low one could have a LPF with a longer time constant to reduce power consumption.

The stored parameters are then used while compiling the final design. In general, this would be done as a part of a larger system, since a shift register could be used to observe intermediate points. A block diagram of the compiled final design is shown in Fig. 16(a). The on-chip processor controls the shift register, a 14-bit ramp ADC and a DAC while executing the algorithm. Fig. 16(a) also shows the DUT with vectorized outputs, where N is 12 here. In general, the system can be scaled as needed to a larger number of filters only constrained by the number of CABs in the FPAA. Fig. 16(b) shows an example flow chart of the tuning algorithm.

Initially, the filters are compiled with a low corner frequency bias except for the first filter, which is used as a reference. Again, the parameters of the reference filter can be selected depending on the application. The tuning algorithm reduces the variation of center frequency, quality factor, gain at the center frequency, and the DC offset with respect to the reference filter. The first step after compiling the design is to reduce the DC offset. Without DC offset reduction, the tuning algorithm will have large errors due to the fact that the system is measuring the amplitude and detecting the minimum value of the output to determine the frequency. The DC offset is reduced by programming the input floating gate of the band-pass filter and measuring it at

the output of the LPF. Thus the system can control the offset of the whole chain. The DC offsets of the 12 filters, scanned using the shifter register, can be seen at the top of Fig. 15(c). The maximum variation can be reduced to 32mV from 0.9 V, after using the tuning algorithm, as shown in bottom of Fig. 15(c). After reducing the DC offset, the tuning algorithm reduces the variation in center frequency, quality factor and gain at the center frequency of the filters by measuring them with respect to the reference filter. The reference filter is characterized by measuring its output at F_{low} and F_{high}, by generating a signal at those frequencies with a DAC. Tuning of the rest of the filter bank is done in two steps, first tuning the high-frequency pole and then the low-frequency pole. Injection is performed, as discussed earlier, to vary the biases to change the feed-forward G_{m2} and the feedback G_{m1}. The filters are injected until they are around an acceptable error from the value of the reference filter. Figure 15(b) shows the frequency response of tuned filters. The variation in filter bank parameters is shown in Fig. 15(d). Specifications of the DUT are shown in Table 2. A set of 12 parallel band-pass filters, amplitude detectors and LPFs consumes power of 7.072μW. Area is reported in terms of number of CABs used by the compiled design, since that is more relevant while designing on an FPAA.

Table 2: Specification of the DUT system

Parameter	Values
Area	12 CABs
Power of 12 filter banks	7.072μW
Technology	350nm
Power Supply	2.5V

3.4 Algorithm over different FPAA

Open-source high-level tools [29], the programming algorithm [10] and the calibration of chip-to-chip variation in the programming infrastructure [42] enables compiling the same design on different FPAAs. The design was compiled on three different chips,

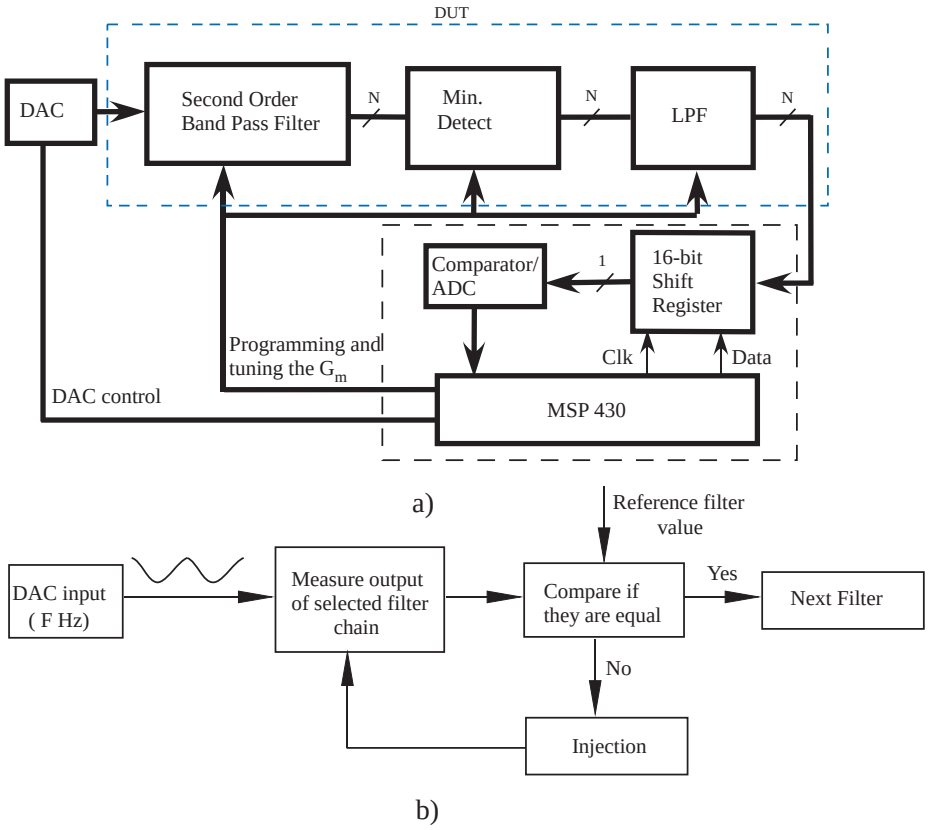


Figure 16: (a) Block diagram of the compiled design. The DUT used for the experiment is shown with vectorized interconnects, where for this work N is 12. (b) A flow chart of the tuning algorithm. The output of the shift register is measured using a 14-bit ramp ADC and stored in the available data memory. A 7-bit DAC controlled by the microprocessor is used for generating a sine wave, at a desired frequency (F Hz).

which were fabricated on the same wafer, to measure the variation and mismatch. This also allows us to test the portability of the built-in self-test system. Table 3 shows the variation in parameters when compiled and programmed with the same current value, before using the tuning algorithm. These values are deviation of each parameter from its mean and then an average of these deviation was taken over three chips. The table in Fig. 17 shows the variation in parameters for each FPAA.

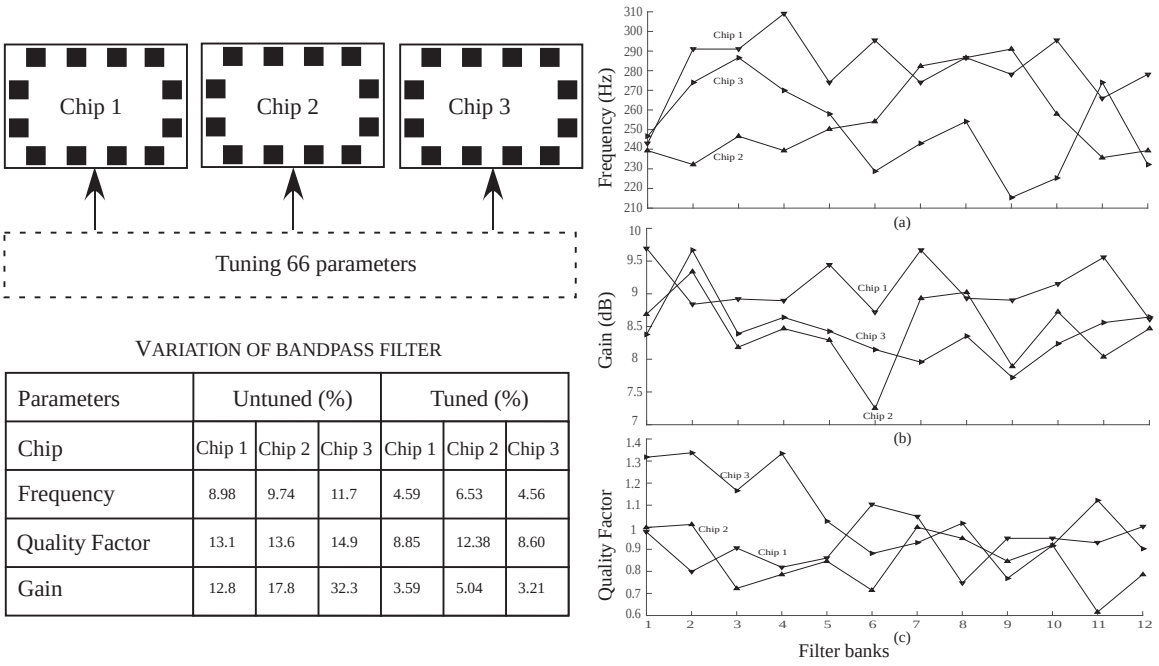
The same algorithm was applied to filter banks compiled on all three chips. The procedure discussed in Section 3.3 was followed, with the filter in the first CAB serving as a reference filter. The bandwidth of the amplitude detectors and the time constants of LPFs were tuned first and then the DC offsets of DUT chain were tuned. Figure 17 depicts the tuning of 66 parameters for three FPAA ICs using the algorithm. The table in Fig. 17 shows the percentage deviation of parameters from their means. In Fig. 17(a),(b) and (c), absolute variation of center frequency, gain at the center frequency and quality factor, after tuning, is plotted. Table 3 shows the deviation of each filter parameters from its mean, as an average over all three chips.

Table 3: Deviation of the values from its mean

Parameter	Untuned	Tuned
Frequency Variation	10.16%	5.23%
Quality Factor Variation	13.86%	9.94%
Gain Variation	21.04%	3.95%
DC offset Variation	0.95 V	29 mV

3.5 Summary Discussion and Comparison

Anon-chip BIST system was presented for the FPAA fabricated in a 350nm CMOS process. The DUT consisted of a bank of 12 filter chains, commonly used for real-time signal processing applications and in large neuromorphic systems. The algorithm was tested on three different chips to test its performance and portability of the BIST system. The variation of parameters for each chip is shown in Fig. 17. The proposed



VARIATION OF BANDPASS FILTER

Parameters	Untuned (%)			Tuned (%)		
Chip	Chip 1	Chip 2	Chip 3	Chip 1	Chip 2	Chip 3
Frequency	8.98	9.74	11.7	4.59	6.53	4.56
Quality Factor	13.1	13.6	14.9	8.85	12.38	8.60
Gain	12.8	17.8	32.3	3.59	5.04	3.21

Figure 17: Tuning for three different FPAA chips using the algorithm proposed in this Chapter. Table shows the values for variation in parameters before and after tuning, for each chip. (a) Variation in center frequency for 12 filters over three different chips. (b) Variation in gain at the center frequency. (c) Variation in quality factor.

Table 4: Comparison of low-power continuous-time filters

Ref	Process	Tuning Parameters	Power Consumption (Normalized per filter)	Design
This system	350nm	66	152.25nW (200 Hz)	Fully reconfigurable
[44]	0.8μm	2 (F_c and Q)	2.5μW (100 Hz)	Custom design with programmable bias
[45]	1.2μm	1 (F_c)	27.86μW (0.83 Hz)	Fully custom design
[46]	350nm	32 (F_c and A_v or F_c and Q)	198nW (200 Hz)	Custom design with programmable bias
[47]	350nm	3 (F_c, A_v and Q)	290nW (30 Hz)	Fully Custom design

system also tunes the bandwidth of the amplitude detectors and the time constant of the LPFs, which are critical parts of the system. The system uses an on-chip DAC and an ADC to generate the necessary signals and to measure the outputs of the filter bank, via a compiled 16-bit shift register. The compiled system consumes 7.072μW of power.

The proposed system automatically tunes multiple parameters to compensate for local mismatch and routing capacitance but does not consider variations in temperature, which will be discussed in the Chapter 4. Variation in power supply is typically controlled by using a precision voltage reference [48]. Power supply rejection ratio of

the C4 structure has been discussed in detail [39]. Previous work has addressed the precision of FG programming [10], FG drift, and charge leakage [49] [50].

A Comparison of this work with other state-of-the-art low-power adaptable continuous time filters is shown in Table 4. The power consumption reported was per filter along with the center frequency of the BPF. The power consumption of the band-pass filter in this work is 152.25nW at 200 Hz. The compiled system and the algorithm tunes 66 parameters. The parameter that are tuned here are center frequency, quality factor, passband gain of BPF. Also, τ of LPFs, DC offset of the filter chain and bandwidth of amplitude detectors are tuned.

The number of parameters available in a FPAA, as with an FPGA, is larger than would typically be in an ASIC because of the reconfigurability of the SoC. The density of parameters, in the case of a FG-based FPAA, is high since the routing infrastructure can also be used as computational and tunable elements, which is not the case in an FPGA, where the routing fabric is normally considered overhead. In general, the number of tunable parameters is restricted by the number available CABs, CLBs, the precision of routing elements, available measurement infrastructure, storage blocks and mismatch caused by capacitance. Also, the number of parameters would be restricted by their orthogonality to each other. The parameters for a custom analog chip is restricted by the density of tunable parameters. If FGs are not being used one approach would be to use small non-linear (in that they do not have to be precise and could have some non-linearity) DACs and calibrating them ahead of time. In [51], such an approach is used for storing the weights of the synapses of a neuron. In [52] a 5-bit DAC is used to supply the bias of G_m-C elements. In the case of reconfigurable analog systems, such as the one used for speech processing [37], the number of parameters which could be tuned are more than 50,000 [25].

CHAPTER IV

MODELS AND TECHNIQUES FOR TEMPERATURE ROBUST SYSTEMS ON A RECONFIGURABLE PLATFORM

4.1 Analog Processing and Temperature Dependence

The number of systems combining elements from within and among the emerging technologies of sensors, communications, and robotics grows every day. The computational abilities of these systems affect the overall system performance through various aspects (e.g., functionality, battery-life, foot-print, etc.). Traditionally, most computational tasks have been performed in the digital domain, which can achieve high-resolution computation at the cost of high power consumption [53]. For systems with a limited power budget, however, low-power real-time computation techniques have been sought after. Accordingly, analog signal processing has been used extensively as an energy-efficient alternative to digital options [54] [55] [56].

The recent mixed-mode large-scale Field-Programmable Analog Array (FPAA) enables advanced functionality for a wide spectrum of sensor applications [25]. Many of the systems that can benefit from the computational power of an FPAA need to operate over a range of environments and temperatures. For instance, modern ubiquitous medical health assessment systems use physiologic signals collected from ambulatory subjects during daily outdoor activities [57]. Likewise, point-of-care diagnostic platforms aiming to achieve lab-quality tests in low-resource settings, operate in environments with varying ambient temperatures [58] [59]. Furthermore, in assisted-living applications, sensor networks are used to identify and track daily activities of the elderly in outdoor settings. For outdoor applications where temperature is not

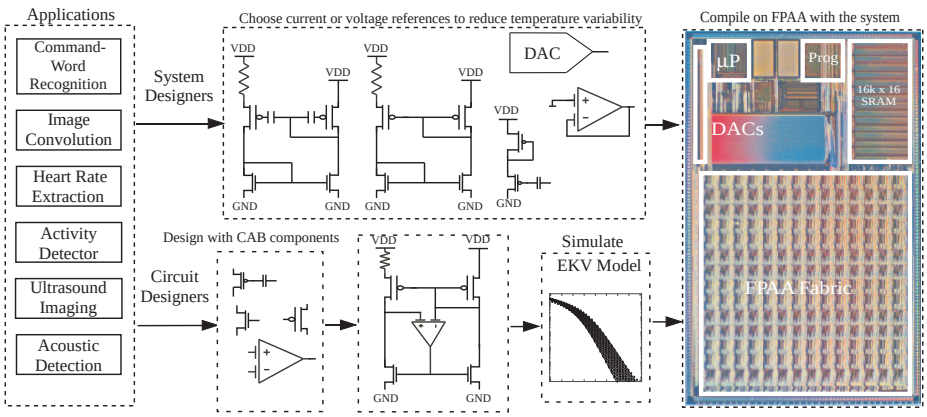


Figure 18: Proposed method to reduce temperature variability while compiling a system for a desired application on an FPAA. The tool infrastructure offers flexibility depending on the user. The tools would compile the desired system along with the selected references on the FPAA. In the case of a circuit designer, the FPAA tool infrastructure enables designing a custom reference circuit. The circuit can be simulated, with models based on EKV before compiling the design on the hardware.

stable, one of the critical performance metrics affecting the computation accuracy of the FPAA would therefore be, robustness against temperature variations.

This Chapter presents a range of techniques and models to estimate and reduce temperature variability of various systems implemented on an FPAA. Figure 18 shows the proposed method a user could follow depending on his or her specific application. A range of voltage and current reference generators are available to be compiled on the FPAA to reduce the temperature variability of the system. Here, we propose to use the FG as a programmable element to achieve reasonable temperature insensitivity rather than trimming/single value FG to achieve precision as shown in [48]. These references form standard blocks in the open source tools built in the Scilab/Xcos environment available online [30] [60]. Here we utilize a simulation model built inside the XCOS tool [60] to model circuits block temperature variation. Temperature measurements were performed using a ZPlus (Cincinnati Sub-Zero Products LLC, Sharonville, OH) temperature chamber. For each temperature value, 15 min. is allowed to ensure that the FPAA die reaches the desired temperature value.

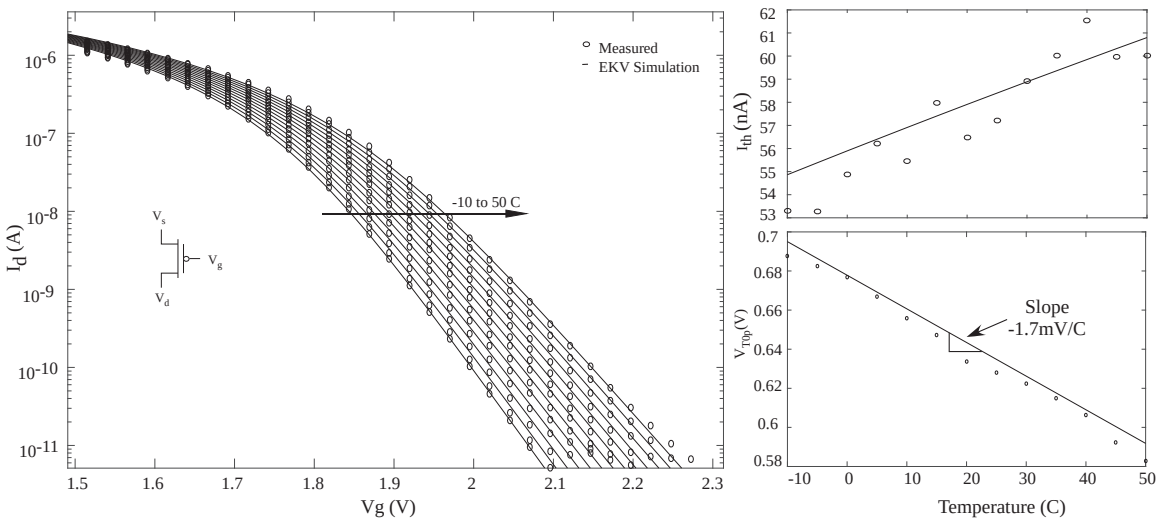


Figure 19: I_d vs V_g transfer characteristics of PMOS over Temperature. EKV modeling is used for modeling the transfer characteristics of a PMOS. I_{th}, current at threshold voltage, and threshold voltage, are also plotted over temperature. The simulated values are consistent with the measured values. The I_{th} and threshold voltages are extracted by curve fitting onto to the output current in both measurement and simulation to be consistent.

Our discussion of the temperature behavior of programmable circuits in the SoC FPAA will proceed in six stages. In Section 4.2, various temperature models for devices available in an FPAA are presented. These models are then used to estimate the temperature variation in simple single-ended circuits in Section 4.3 and these simulations are compared with measured data. In Section 4.4, various references are introduced and their variability with temperature is studied. A bootstrap FG reference generator is introduced in Section 4.4.1. Section 4.4.2 shows resistorless voltage references that can be compiled on the FPAA. Section 4.5 and section 4.6 show the performance of various signal-processing systems and circuits over temperature.

4.2 Modeling Temperature Dependence

A simulation model adapted from the EKV model [61] for all regions of operation is developed. Based on this model, the channel current for a pMOS and an nMOS transistor are governed by the following equations

$$
\begin{aligned}
I_d =& I_{thnmos} \ln^2 \left(1 + e^{(\kappa(V_g - V_{T0n}) - (V_s) + \sigma(V_d))/2U_T}\right) \\
&- I_{thnmos} \ln^2 \left(1 + e^{(\kappa(V_g - V_{T0n}) - (V_d) + \sigma(V_s))/2U_T}\right)
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
I_d =& I_{thpmos} \ln^2 \left(1 + e^{(\kappa(V_{DD} - V_g - V_{T0p}) - (V_{DD} - V_s) + \sigma(V_{DD} - V_d))/2U_T}\right) \\
&- I_{thpmos} \ln^2 \left(1 + e^{(\kappa(V_{DD} - V_g - V_{T0p}) - (V_{DD} - V_d) + \sigma(V_{DD} - V_s))/2U_T}\right)
\end{aligned}
$$

where I_{thnmos} and I_{thpmos} are specific currents at threshold and their dependance on process parameters and temperatures is given by

$$
I_{thnmos} = 2\mu_{nmos} C_{ox} (W/L) U_T^2 / \kappa \tag{5}
$$

and (6)

$$
I_{thpmos} = 2\mu_{pmos} C_{ox} (W/L) U_T^2 / \kappa. \tag{7}
$$

In (4), σ is the drain-induced barrier lowering coefficient, and V_d, V_s, V_{T0p}, V_{T0n}, and U_T are the drain, source, pMOS zero-bias threshold, nMOS zero-bias threshold, and thermal voltages, respectively. The temperature dependence of I_d in (4) arises from the threshold voltages, I_{th}, and U_T. The dependence of the threshold voltages on temperature can be modeled using $A_1 + A_2 U_T$. I_{th} has dependence on temperature due to the mobility (μ) and presence of U_T^2, which can be modeled using $I_{thr}(\frac{T}{T_r})^a$, where $a \approx 0.5$. From the measured data shown in the the Table 5, which shows

$$
\frac{dI_{th}}{dT} / I_{thr} = \frac{a}{T_r}, \tag{8}
$$

it can be seen that $a \approx 0.6$. Here, T_r is the reference temperature (298K). The variation of V_{T0} is as follows

$$
\frac{dV_{t0}}{dT} = -\frac{2}{\kappa} \ln \left(\frac{N_D}{\sqrt{N_c N_v}} \right) \frac{dU_T}{dT} \tag{9}
$$

where N_c and N_v are effective density of electrons and holes in conduction and valence band respectively. Their dependence on temperature is $T^{3/2}$ [62]. This is small

compared to the linear dependence to U_T term. It should be noted that the model in [61] has more number of parameters and is much more generalized. In the case of (4), the model has a reduced number of parameters, which allows for faster simulation, with the ability to closely predict data from the FPAA.

Table 5: Comparison of simulated and measured data: Percentage change over $60°C$.

Device	Measured				Simulated			
Parameter	Threshold Voltage		I_{th}		Threshold Voltage		I_{th}	
pFET	-0.28%	-1.7mV/C	0.2%	2000ppm/C	-0.27%	-1.7mV/C	0.17%	1700ppm/C
nFET	-0.26%	-1.1mV/C	0.24%	2400ppm/C	-0.22%	-0.95mV/C	0.17%	1700ppm/C

This model has been integrated as a part of the Scilab/Xcos FPAA development environment [60]. Figure 19 shows measurements of a pFET compiled on to the FPAA and corresponding simulations performed using the EKV model. The tool incorporates the above threshold variation in U_T, threshold voltage and current at threshold voltage (I_{th}). Figure 19 compares these temperature variations between the simulated model and the measured results over a change of 60°C. The measurements are silicon data obtained from the FPAA fabricated in 350nm technology.

Using the above model for simulation, similar measurements and simulations were performed for an nFET. The variation in the threshold voltage and the current at threshold voltage (I_{th}) for the devices are summarized in the Table 5. These parameters are extracted from the transfer characteristics using a EKV curve fit program in scilab [63], with varying U_T. The variations are shown as percentage changes in the parameters from the values at room temperature over 60°C.

The consistency between the measurements and the model allows us to predict the first-order behavior of circuits and systems compiled on the FPAA with temperature, thereby enabling temperature-robust circuits and system design.

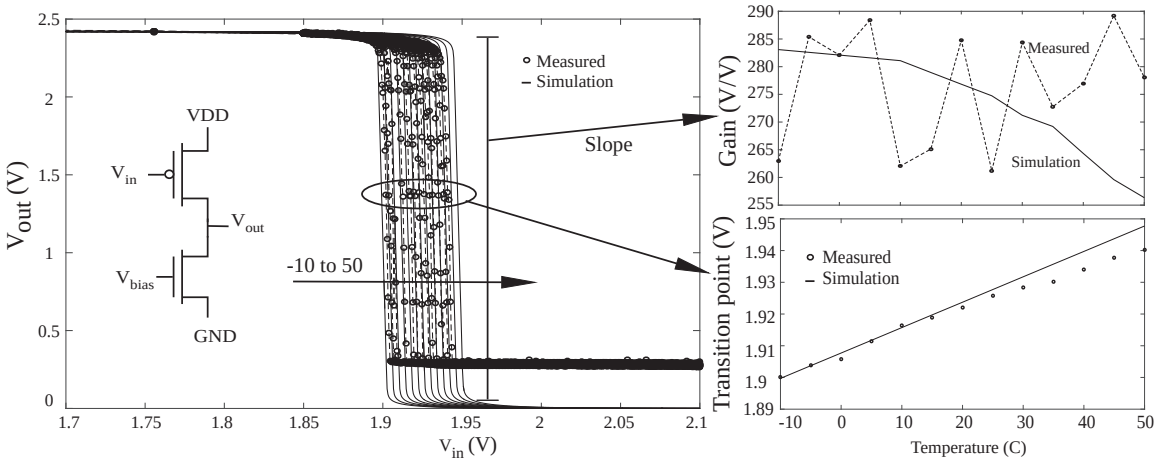


Figure 20: Transfer function of a common-source amplifier, measured and simulated using the models developed in the previous section, over temperature. The slope, and hence the gain of the common source amplifier is also plotted. The slope is constant over temperature. The transition of V_{out} with V_{in} for different temperatures is also shown. The transition changes over temperature because the threshold voltages of nMOS and pMOS transistor vary differently.

4.3 Temperature Dependence of Simple Single-Ended Circuits

The models developed in the previous section predict the behaviour of circuits and systems on the FPAA. To illustrate, a common-source amplifier, shown in Fig. 20, has a gain which is constant over temperature [64]. The EKV model shown in (4) can be reduced to following set of equations when $I_{sat} \ll I_{th}$:

$$I_d = I_{thnmos} e^{(\kappa(V_g - V_{T0n}) - V_s + \sigma_{nmos} V_d)/U_T} \text{and}$$

$$I_d = I_{thpmos} e^{(\kappa(V_{DD} - V_g - V_{T0p}) - (V_{DD} - V_s) + \sigma_{pmos} V_d)/U_T}.$$

Equating the channel current for nMOS and pMOS we have

$$I_{thnmos} e^{(\kappa(V_{bias} - V_{T0n}) + \sigma_{nmos} V_{out})/U_T} = I_{thpmos} e^{(\kappa(V_{DD} - V_{in} - V_{T0p}) + \sigma_{pmos}(V_{DD} - V_{out}))/U_T}$$

and

$$V_{out} = \frac{-\kappa}{\sigma_{nmos} + \sigma_{pmos}} V_{in} + V_{offset}. \tag{10}$$

In (10), the offset V_{offset} is a manifestation of U_T, the threshold voltage difference between the nFET (V_{T0n}) and the pFET (V_{T0p}), and $log(\frac{I_{thnmos}}{I_{thpmos}})$. In (10), κ for pFET and nFET have been taken to be equal for the ease of calculation but typically these values are different from each other. Fig. 20 shows the transfer characteristics of a common-source amplifier, measured on the FPAA and simulated using the models developed before, with a pFET input. As seen in Fig. 20, the slope of the simulation and measurement remains relatively constant. The variation in the transition points of the transfer characteristics corresponding to different temperatures is associated with the offset (V_{offset}) term, which has a dependence on U_T and also due to the fact that the threshold voltage of the pFET and the nFET vary differently over temperature.

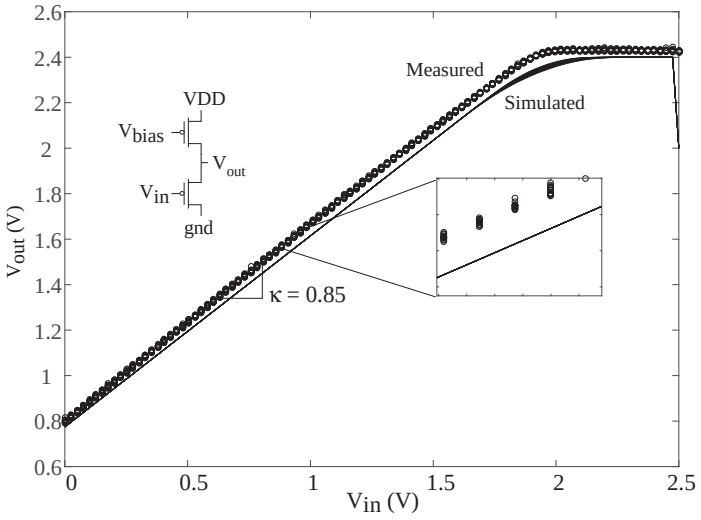


Figure 21: Transfer function of a common-drain amplifier, measured and simulated using the models developed in the previous section, over temperature. The slope and the transition offset of the circuit is constant with temperature because the variation in threshold voltage of the two pMOS devices are similar. A slope of 0.85, which is the κ of the pMOS device, is measured from the circuit compiled on the FPAA.

For temperature-robust transition points the circuit could be designed to utilize the same type of FET devices. The transfer characteristics of the two pFET based common drain is given by

$$I_{thpmos}e^{(\kappa(V_{DD}-V_{bias}-V_{T0p})+\sigma_{pmos}(V_{DD}-V_{out}))/U_T} = I_{thpmos}e^{(\kappa(V_{DD}-V_{in}-V_{T0p})-(V_{DD}-V_{out}))/U_T}$$

and

$$V_{out} = \frac{\kappa}{1+\sigma_{pmos}}V_{in} - \frac{\kappa V_{bias}}{\sigma_{pmos}+1} + V_{DD}. \tag{11}$$

Thus, the slope of the transfer characteristics is $\approx \kappa$, since $\sigma_{pmos} << 1$, which is invariant over temperature. Also, the V_{offset} term is independent of U_T, threshold voltage, and I_{thpmos}, which makes the transition points invariant with temperature. Figure 21 shows the transfer characteristics of a common drain circuit with a pFET input. The measurement and simulation are plotted and have similar slope, that is the κ of the pFET input. The inset shows a zoom-in of the transfer characteristics measured over $60°C$.

4.4 Reducing Temperature Dependence in Programmable Circuits and System

The programming and biasing of an FG switch or an FG current source is generally done using a DAC. During programming, when the circuits and systems are getting compiled on the FPAA, the DAC allows us to vary the bias on the gate in order to compensate for variation and mismatch on the FPAA [42]. It is typically assumed that the temperature will not vary drastically while programming the device. In run mode, however, when the implemented system is used for computation and processing on the FPAA, the temperature can vary based on the application; for example, in wearable systems [55, 56, 65] or as a sensory node for analyzing speech [56]. A subthreshold current reference circuit, similar to a conventional current reference circuit [66], shown in the Fig. 22 could bias the gate of a pFET, which would reduce the temperature variation of the output current to 12.0% over $60°C$. The bias current is constant and is not programmable unless the $\frac{W}{L}$ of its transistor and the resistor values are changed. In the case of an FPAA, it would be possible to change $\frac{W}{L}$ by adding pFETs

and nFETs in parallel. However, that would mean adding extra capacitance because of the routing between different CABs.

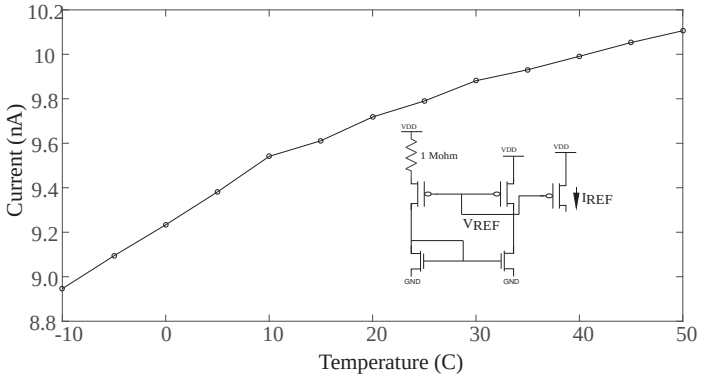


Figure 22: A bias reference circuit compiled in the FPAA. The circuit uses multiple pFETs to create the required β multiplication. This is a drawback, since it has to use multiple CABs.

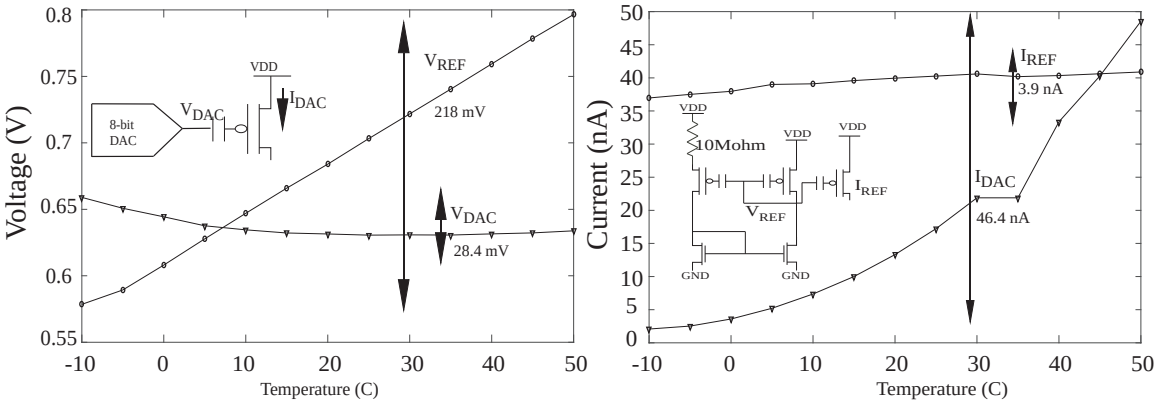


Figure 23: A FG-based bootstrap reference circuit compiled on the FPAA for biasing the FG transistors. The DAC, which is generally used to bias the FG transistor, is also characterized over temperature. Output of the reference generator (V_{REF}) is plotted over a temperature variation of $60°C$. The output of the DAC is also measured over the same temperature range. Also, the effects of temperature on the drain current is studied by biasing a FG transistor in both modes; i.e., bias using a DAC and the bootstrap reference circuit.

4.4.1 FG-Based Reference Circuit in Subthreshold

The reference circuit shown in Fig. 23 is based on the bootstrap reference architecture [67]. However, unlike the conventional bootstrap reference architecture where

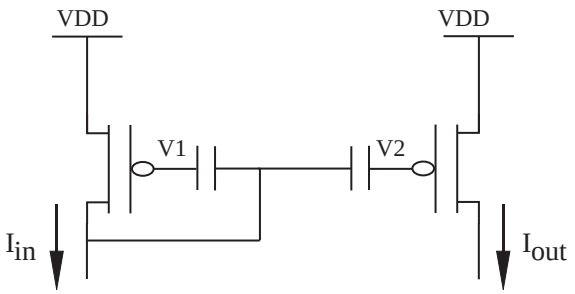


Figure 24: A simple FG pFET based current mirror. The mismatch between the two transistor causes I_{out} to be not equal to I_{in}. This also results in temperature dependence of I_{out} assuming I_{in} is kept constant.

the difference between the aspect ratios of pFETs is responsible for the reference generation, the V_{REF} in Fig. 23 is generated because of the difference in the amount of charge stored on the FG pFETs. This bootstrap reference circuit is compiled using FG pFETs from the switch fabric, which are part of the local interconnect routing present in the CAB, and an nFET current mirror, which is a part of the CAB elements described earlier.

Figure 23 also shows measurements of V_{REF}, I_{REF}, V_{DAC}, and I_{DAC} over temperature. A variation of 28.4 mV was observed in the DAC, which translates into a temperature variation of 189 ppm/°C with a linear range of 2.5 V. This manifests as a large change in current, when used to bias a FG, since the voltage does not scale with temperature to compensate the variation in threshold voltage and I_{th}. The bootstrap current reference has a variation of 218 mV in V_{REF} and the current mirror output has a temperature drift of only 3.9 nA.

Evaluating the relationship of I_{out} and I_{in} over temperature of a current mirror built with a FG pFET, such as the one used in Fig. 23 for biasing and shown in the Fig. 24, is given by

$$I_{out} = I_{in} e^{\kappa(V_1 - V_2)/U_T} \tag{12}$$

If V_1 and V_2 in the above equation are equal, then we have $I_{out} = I_{in}$. In general, for a

non-FG current mirror, there is a threshold voltage mismatch between the transistors that will lead to a variation of I_{out} with temperature. The variation of current in this case can be modeled as

$$I_{out} = I_{in}A = I_{in}A_0^{T_0/T}, \tag{13}$$

where $A = e^{\kappa(V_1-V_2)/U_T}$ and T_0 is a reference temperature which could be room temperature for simplicity. This effect is important in the case of VMM, discussed in Section 4.5, where multiple FG pFETs are biased as a current mirror.

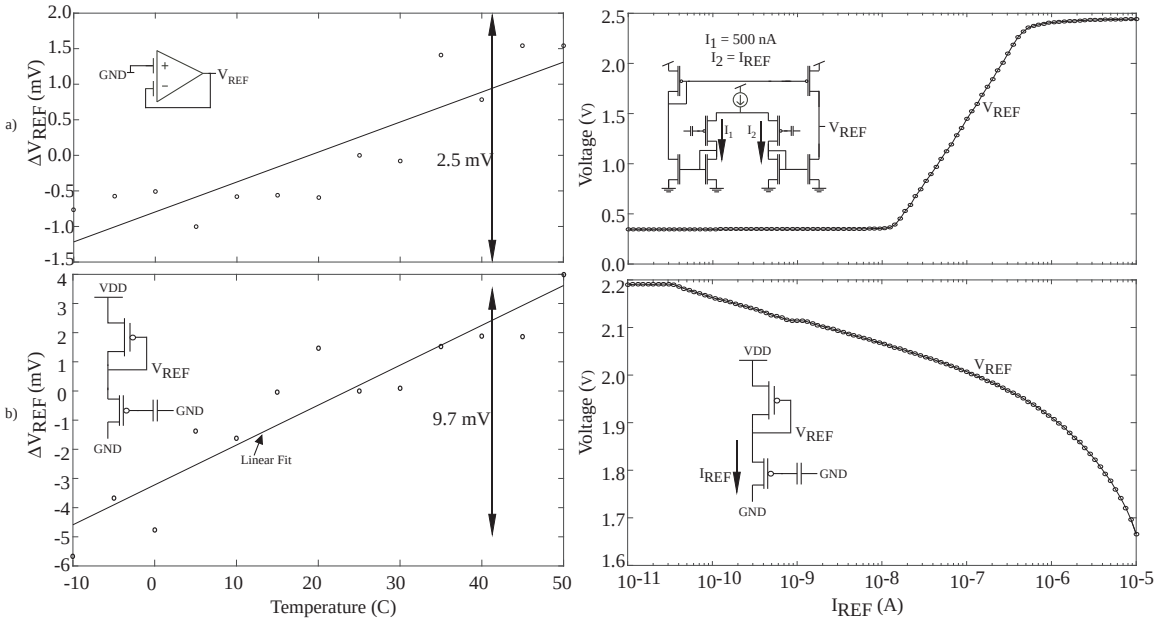


Figure 25: Programmable voltage reference without resistors. a) The figure on the top shows results from a voltage reference designed using an FGOTA connected in a follower configuration. V_{REF} is generated by creating an offset between the positive and negative terminals of the FGOTA. A variation of 2.5 mV was observed in the reference voltage value over a change of 60 $°C$. The reference has a programmable range of 0.346 V to 2.441 V, which is shown in the second part of the figure. b) A programmable voltage reference designed using a diode connected pFET and a FG pFET from the routing infrastructure. A variation of 9.7 mV was observed for a change of 60 $°C$. The range of this reference is 1.66 V to 2.19 V.

4.4.2 FG-Based voltage reference without resistors

The main drawback of the reference circuits introduced in the previous section is the need for an external resistor for different bias currents. There are several resistorless bias reference circuits that take advantage of availability of different V_{tnmos} or V_{tpmos} device,(i.e high and low threshold devices) [68] [69]. The aforementioned techniques require an ASIC and availability of different threshold devices in the process.

Here we propose to use FGs to create differences in threshold voltage by programming different charges at the floating nodes. This allows us to bias the reference with different biasing values and to be able to directly compile the circuit onto the FPAA. Figure 25 shows the schematics of the proposed reference circuits. Figure 25a introduces a voltage reference circuit using an FGOTA. V_{REF} is generated by creating an offset between positive and negative terminals of the FGOTA. A $\triangle V_{REF}$ of 2.5 mV, with respect to V_{REF} at $25°C$, was observed over a range of $60°C$. The range of voltage reference that can be generated with this reference is between 0.346 V to 2.44 V and is plotted in Fig. 25a. It should be noted that, V_{REF} is constrained near the GND by the output buffer used for the measurement and not due to the linear range of the FGOTA. Here, I_{REF} is the programming current, and therefore an indicator of the charge programmed onto the FG, which is varied to tune V_{REF}. For the measurements in Fig. 25a, the FGOTA was biased at $2\mu A$, and hence the power dissipated by the circuit is 5 μW. This could change depending on the application,(i.e the load of the reference, and the power requirements. Considering the above linear range of the FGOTA-based voltage reference, it has a temperature variation of 19.83 $ppm/°C$.

Fig. 25b shows a voltage reference built using two pMOS transistors. The FG transistor used in the circuit is part of the routing fabric in the FPAA whereas the pMOS transistor is part of the CAB. A variation of 9.7 mV over $60°C$ was observed in the case of this voltage reference with programming range of 1.66 V to 2.19 V. Considering this linear range, the reference has a temperature variability of 305 $ppm/°C$.

The dependence of V_{REF} in the case of Fig. 25b can be analyzed by reducing the EKV equation in (4). As seen in (14), V_{REF} has a weak dependence on U_T and supply voltage VDD. In the following derivation, it is assumed that the κ of the FG pMOS and the diode connected pMOS is similar and that their $\sigma \approx 0$:

$$I_{thpmos}e^{(\kappa(V_{DD}-V_{REF}-V_{T0p})/U_T} = I_{thfgpmos}e^{(\kappa(V_{DD}-V_{fg}-V_{T0p})-(V_{DD}-V_{REF}))/U_T}$$

$$U_T ln(\frac{I_{thpmos}}{I_{thfgpmos}}) = \kappa(V_{REF}-V_{fg}) - (V_{DD}-V_{REF})$$

$$V_{REF} = \frac{U_T}{\kappa+1} ln(\frac{I_{thpmos}}{I_{thfgpmos}}) + (\frac{\kappa}{\kappa+1})V_{fg} + (\frac{V_{DD}}{\kappa+1}) \quad (14)$$

It should be noted from the above equation that if the I_{th} of two devices are equal, the output V_{REF} would be almost invariant over temperature. In addition, the above analysis holds only if the transistors are in the subthreshold saturation regime. Also, the two-transistor circuit would have fairly significant power-supply coupling as opposed to the FGOTA whose PSRR will be high.

4.5 Vector Matrix Multiplication

A FG-based Vector Matrix Multiplication (VMM) current is one of the building blocks in an analog classifier [37, 70]. It stores the weights of the classifier as charges (Q) on FG pFETs. The output of a VMM is a current signal, which is the product of input voltage and the weight stored on the floating node. The equation in (4) can be adapted to describe the VMM as follows:

$$\begin{aligned} I_d &= I_{th}e^{\left(\kappa_{fgpmos}(V_{dd}-V_{fg}-V_{T0p})/U_T\right)}e^{-(V_{dd}-V_{in})/U_T} \quad (15)\\ I_d &= I_{th}e^{\left(\kappa_{fgpmos}(V_{dd}-V_{fgref}-V_{T0p})\right)/U_T}We^{-(V_{dd}-V_{in})/U_T} \\ W &= e^{\left(\kappa_{fgpmos}(-\Delta V'_{fg})\right)/U_T} \end{aligned}$$

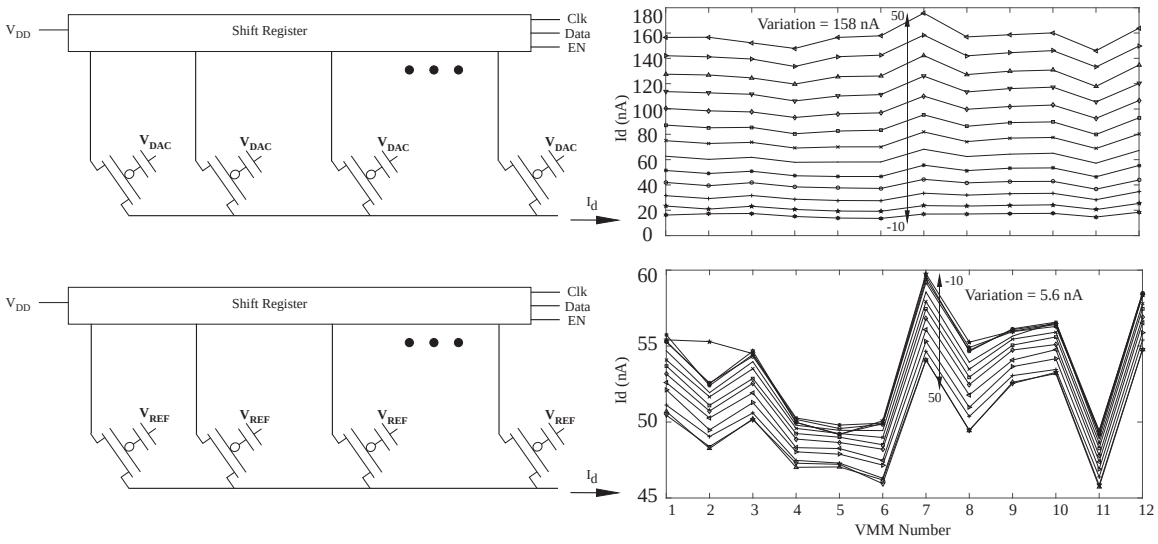


Figure 26: A typical implementation of a Vector Matrix Multiplication, shown here with a shift register for measurement, in an FG based FPAA. The two measurement corresponds to two different experimental setups: one where a DAC is used to bias the FG whereas the other corresponds to a FG biased using the voltage reference discussed earlier. A current variation of 158 nA was observed over $60°C$ when the VMM, which has been programmed to get 50 nA at room temperature, was biased using a DAC. A variation of 5.6 nA was observed in case of the VMM biased using the voltage reference.

The above set of equation reveals dependence of weight on temperature. Also, V_{in} would have a different relationship with the output I_d with a change in temperature given by $e^{\frac{-(V_{dd}-V_{in})}{U_T}}$. Thus, the classifier, instead of having an output $y = W' * x$, will have a temperature-dependant term.

To demonstrate the dependence of the VMM on temperature, when biased with a DAC, a 12×1 (i.e. a 12 inputs and 1 output) VMM is compiled on the FPAA. As shown in Fig. 26, a shift register is used to measure each FG output over $60°C$. A variation of 158 nA is observed when they are nominally biased around 50 nA at room temperature. This variation could lead to substantial errors while using the VMM in classification as temperature could vary depending on the application. As an example, when the classifier is used for classifying acoustic signal from the knee joint [65], the system temperature is expected to change with the environment

temperature (e.g., room temperature during activities performed at home or hot/cold temperatures during outdoor activities). To address this issue, the current reference introduced in the Section 4.4.1 is used to bias the FG pMOS of the VMM. Figure 26 shows the variation of the VMM current when biased using the FG current reference. A maximum variation of 5.6 nA is observed in the current. This variation is due to the fact that there is mismatch in the threshold voltage of the transistor resulting in $A_0 \neq 1$, where A_0 is the gain factor from (13).

The VMM weight is updated depending on the learning of the classifier, and hence, in practice, the weight will have certain $\triangle V_{fg}$ compared to the biasing circuit. This will result in a gain $A_0 \neq 1$, thereby leading to dependence of I_{out} on temperature. If the output stage of the VMM is a Winner Take All (WTA) [71], which is the case in [55, 65, 70], the small variation would not affect the output of the classifier. This is due to the fact that a WTA compares relative current between the competing branches of the VMM.

4.6 Temperature Variation of a Band-Pass Filter

A $G_m - C$-based second order band-pass filter is used extensively for several signal processing systems [55, 56, 65]. A second-order $G_m - C$ filter enables extracting frequency-based features from an input signal with a low power consumption. Also, a FG-based $G_m - C$ offers programmability over broad range of center frequencies by changing the bias of the FGOTAs. The input of the FGOTA is capacitively coupled to increase the linearity of the filter. The PSRR, CMRR, input referred noise, and other characteristics of the filter are described in detail [39].

The inset in Fig. 27 shows the schematic of a $G_m - C$ filter compiled in a single CAB of the FPAA. The FGOTAs are biased using a FG pFET, as shown in Fig. 27. Fig. 27 shows the frequency response of the band-pass filter over 60°C of temperature change when biased using a DAC. Figure 27 also shows variation in the center

frequency, quality factor, and gain of the band-pass filter. The variation in center frequency is 2kHz for a center frequency of 840 Hz at room temperature. This variation would lead to significant error in the processing and extraction of features in different signal processing systems.

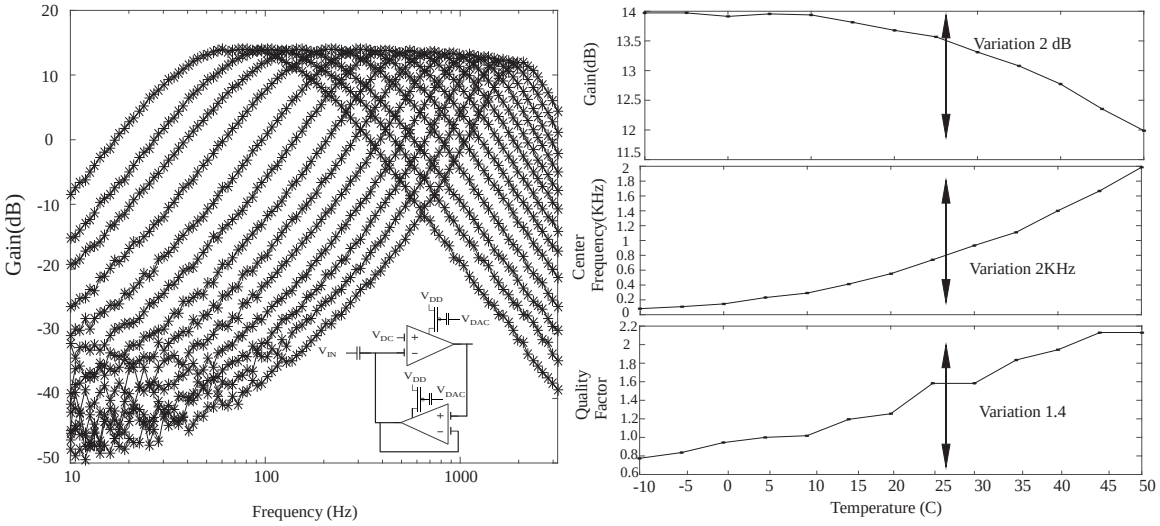


Figure 27: Measured frequency response of a second order band-pass $G_m - C$ filter over temperature. A FG pFET is used as a bias transistor for the OTA. Here, the gate of the FG pFET is biased using a DAC as seen in the inset of the frequency response graph. A variation of 2kHz was observed in the center frequency of the filter. A variation of 2 dB in gain and of 1.4 in the quality factor of the filter was observed.

A FG-based current reference generator introduced in Section 4.4.1 would be suitable for biasing the $G_m - C$ filters. Fig. 28 shows the bootstrap reference compiled along with the band-pass filter. Fig. 28 also shows the frequency response of the band-pass filter over a temperature variation of $60°C$. The PTAT response of the FG-based bootstrap reference helps in compensating the CTAT variation in V_T. Fig. 28 also shows the variation in the characteristics of the band-pass filter: quality factor (Q), center frequency (F_C), and gain at the center frequency. As compared to the case where it was biased by a DAC, the variation in the center frequency is reduced to 76 Hz, gain variation is 1 dB as opposed to 2 dB and quality factor variation is

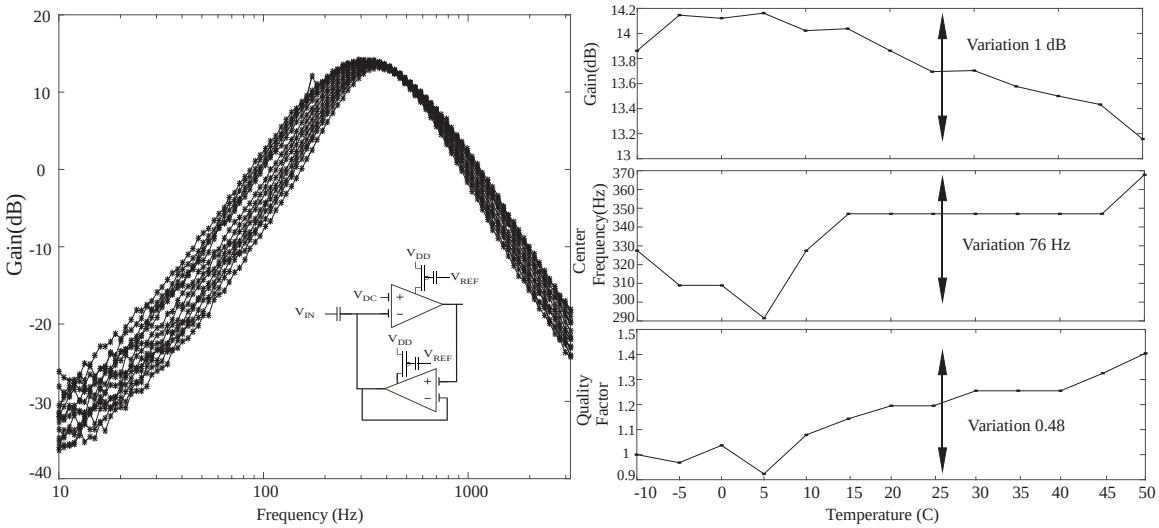


Figure 28: Measured frequency response of a second order band-pass $G_m - C$ filter over temperature. Here, the gate of the FG pFET is bias using the FG based voltage reference. A variation of 76 Hz was observed in case of the center frequency. A variation of 0.48 in quality factor and 1 dB in gain, at the center frequency, of the filter was observed.

0.48 compared to 1.4. The measurements in Fig. 28 and Fig. 27 were performed consecutively, where the filter were biased first with a DAC and then with a FG current reference to keep other variations constant. The resulting temperature dependant variation of 76 Hz observed in the Fig. 28, can be explained due to the fabrication related device-mismatch between the FG pMOS used in the reference circuit and the FG pMOS used as the biasing transistor of the FGOTAs in the band-pass filter. In addition, there are design-related mismatches in this FG pMOS devices as well. For instance, the coupling capacitor of the FG pMOS in the reference circuit is 8 fF as opposed to 43 fF in the FGOTA bias device. The other source of mismatch is the different size of the FG pMOS transistors, which are $\frac{1.8\mu m}{600nm}$ in the FG reference circuit and $\frac{6\mu m}{2\mu m}$ in the FGota. In subsequent designs, of the FPAA, these variations will be reduced by keeping a single size of FG transistors for all the devices on the SoC.

4.7 Summary and Discussion

There has been a growing interest in using FPAAs for rapidly prototyping mixed-signal systems, performing hyprid and analog signal processing for a wide ranging applications, and using programmability and reconfigurability to increase system performance and energy efficiency [15] [72] [73] [25]. Hence, it is important to study the effects of temperature on a reconfigurable platform and investigate methods that can reduce these variations. A FG temperature compensation structure as part of the CAB is used to reduce the variation in current over temperature [74]. This work presents several circuits, models, and techniques to estimate and to reduce the temperature variation of such systems.

The simulations performed using the model developed based on the EKV model were in close agreement with the measured data. These models created for simple nFET and pFET devices were then used to study temperature behaviour of simple single-ended circuits and compared with measurement results. The intuition created here led to building and designing several current and voltage references introduced in subsequent sections. The references developed in this work are a part of the Scilab/Xcos environment as blocks thereby enabling easy compilation as a part of a larger system.

A bootstrap current reference is introduced to bias FG devices on the FPAA. The performance of the bootstrap reference is studied over temperature and measured results from the FPAA are presented. The FG reference circuit is also used to bias two critical components of analog signal processing chain: the VMM and the second-order band-pass filer. Their performance over temperature when biased using a DAC and the FG reference circuit is studied. A variation of 156 nA was observed in the output current of the VMM when biased with a DAC as opposed to a variation of 5.6 nA when biased with FG current reference circuit. The band-pass filters center frequency varies by 2 kHz when biased with a DAC whereas in case of FG reference

circuit it varies by 76 Hz.

This work also presents two resistorless voltage references with a wide range of programmable voltages. The FGOTA-based voltage reference has a temperature variability of $19.83 ppm/^{\circ}C$ whereas the 2-transistor voltage reference achieves a temperature variation of $305 ppm/^{\circ}C$. An FGOTA can also be used as an error amplifier in an LDO without having to generate a separate reference voltage [75]. All measurements were performed on the FPAA, which is powered using the USB V_BUS power supply. PSRR measurements were not performed since body of all pFET devices are connected to the fixed USB supply, thus preventing accurate PSRR measurements.

CHAPTER V

ANALOG PROCESSING FOR BIOSIGNALS

The use of analog signal processing for analyzing and processing physiological signal is highly attractive. The output of various vital-signs monitoring sensors are in the analog domain. As the health-care system shifts focus towards prevention, monitoring physiological signals becomes essential. Continuous data logging and output sampling over the course of a few hours would result in significant power consumption and would require the users to continually charge their devices. That would be cumbersome for the user of wearable medical devices. Thus, there is a need to develop new systems that requires significantly less power, allowing the devices to log and analyze data for several hours to days.

This Chapter introduces several circuits and system that analyze output of various sensors. Eventually, such a system could be used in real-time for continuous monitoring and for providing feedback to the health-care providers.

5.1 Real-Time Vital-Sign Monitoring in the Physical Domain

Heart disease (HD) is the primary cause of death in the U.S. and health-care expenditure for HD represents the largest portion of the total national health spending [76]. Furthermore, it is expected that, by 2035, the percentage of the U.S. population having at least one HD occurrence will rise to 45%. Early diagnosis and timely management of HD can potentially lower risk of complications, thus improving quality of life [76–79]. However the current HD diagnosis approach is reactive and, therefore, inappropriate for early diagnosis: Only after symptoms occur do patients visit the clinic, where expensive procedures are used for diagnosis. By contrast, a proactive

approach in which people are ubiquitously monitored at home for the timely detection of signs of abnormalities before serious symptoms manifest, could increase the rate of early diagnosis. Such a proactive approach could also facilitate treatment because the system is tuned to changes in patients physiology, thereby potentially increasing treatment success rates and reducing the frequency of visits to the clinic as well as the associated health-care costs.

Datasets collected from human-subject studies have shown that several critical hemodynamics and vital-sign variables can be extracted from electrocardiography (ECG) [80], blood pressure (BP) [81], and photoplethysmography (PPG) [82] signals. Accordingly, several research groups and companies have developed wearable/implantable systems to capture and analyze those signals for cardiovascular health assessment outside the clinic [83–89]. The majority of these devices consist of a (network of) sensor(s) collecting analog physiological data, which is digitized by a microprocessor for detailed offline analysis. Unfortunately, a microprocessor consumes milliamperes of current and due to battery-life considerations, it is impractical to only compute using a microprocessor in sensor-nodes distributed across the body. Accordingly, a central processing unit with high computational power might receive and analyze the physiological data collected at different nodes. In such a scenario however, transmission of large dataset to the central processor necessitates high-throughput, and thus potentially power-hungry communication channels.

A potential solution to the aforementioned problem would be localized energy-efficient computation nodes. In a network with distributed and low-power computational capabilities, salient information could be extracted from the raw signals directly at the sensor site, thus significantly alleviating the communication and computational burdens on the central signal processor. As such, a potentially significant energy savings of the overall system can be expected.

In the literature, several studies have reported on distributed and low-power processing of cardiac signals with a common theme of performing computations in the analog domain, thus improving the power-efficiency by as large as three orders of magnitude [90, 91]. More recently, we have demonstrated noise-resistant analog detection of foot-points of impedance plethysmography and PPG waveforms for monitoring blood flow and pulse timing in real time [92, 93].

In this section, we present energy-efficient circuitry performing analog computation for analyzing cardiac signals in real time. One of the distinguishing features of this work compared to other studies on analog signal processing of cardiac signals is the use of FG MOS devices as analog memory elements for tailoring the design with patient-dependent parameters and potentially reducing calculation errors due to device mismatch. For energy efficiency and reduced design complexity, the circuit leverages the CMOS subthreshold region of operation. The design has been implemented on a FPAA platform achieving programmability and reconfigurability through a FG CMOS fabric.

5.1.1 Overview of the Physiological Signals and Features Critical for Vital Sign Calculation

The processor computes vital signs from ECG, BP, and PPG signals (Fig. 29), which are quasi-periodic cardiac signals that can be non-invasively measured using wearable biomedical devices. In an ECG signal, the R-wave, which is the global maxima point in a heartbeat, is critical for calculating the R-R distance, and therefore heart rate. In an arterial blood pressure (ABP) waveform, the maxima and minima points correspond to the systolic (SBP) and diastolic (DBP) blood pressure, respectively. Oxygen saturation (SpO2) can be calculated from two PPG signals at different wavelengths by calculating their perfusion index ratio. In this study, we use two commonly used wavelength PPG signals, red and infrared. The perfusion index of a PPG signal is calculated by normalizing the peak-to-peak variation by the DC level.

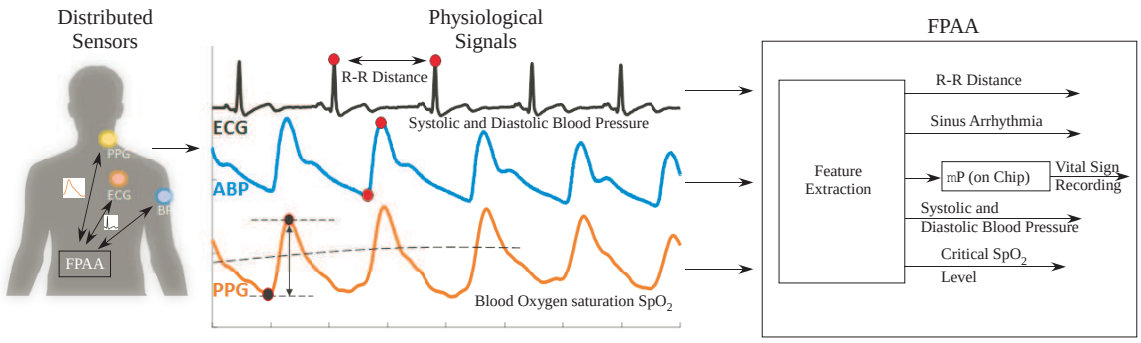


Figure 29: Physiological signals and their analysis on the Reconfigurable Cardiac Processor. Electrocardiogram (ECG), arterial blood pressure (ABP), and photoplethysmography (PPG) with important features and vital information that can be extracted from each signal. Analog processing enables computation without using an ADC and, hence, lower power. This is significant when such devices are used for continuous monitoring. The block diagram of the mixed-signal cardiac signal processor is designed and implemented on an FPAA. The feature extraction and abnormality detection are performed in the analog domain, whereas vital sign value calculation is performed using the on-chip microprocessor.

5.1.2 Methods for Cardiac Signal Processor Validation

5.1.2.1 Physiological Dataset

The accurate detection of R-R distance, SBP and DBP, and SpO2 by the processor has been initially validated using datasets collected from three healthy adults. The ECG dataset was collected using a wireless BioNomadix ECG module (Biopac Systems Inc., Goleta, CA). The BP dataset was collected using a Finapres device (Finapres Medical Systems, Amsterdam, Netherlands) using the volume clamping technique for continuous BP monitoring in a non-invasive manner. Both ECG and BP datasets were digitized using an MP150 biosignal data acquisition system (Biopac Systems Inc., Goleta, CA). The red and IR PPG dataset was collected using an AFE4400SPO_2EVM pulse-oximeter evaluation module (Texas Instruments, Dallas, TX). All signals were collected at a rate of 2kSps per channel. The measurement protocol consisted of rest, handgrip exercise, and deep breathing phases. As such, variations in HR and BP could be obtained. During measurements, the subjects were in a standing position.

The measurements were approved by the Georgia Institute of Technology Institutional Review Board (IRB).

The PPG dataset from the healthy subjects has also been used to test the SpO_2 warning generation circuitry. For validation of the sinus arrhythmia warning generation circuitry, an ECG dataset of an adult with an abnormal heart rhythm has been used [94, 95].

5.1.2.2 Cardiac Signal Processor Validation Setup

For validation purposes, the dataset have been streamed onto an FPAA using the 14-bit DAC in Digilent Analog Discovery. In an ultimate BSN cardiac sensing node scenario, sensors could be directly interfaced with the FPAA using either a low-noise amplifier or capacitively coupled amplifiers, depending on the type of the sensor/transducer being used. Measurements from the cardiac signal processor have been compared with digital signal processing algorithms implemented in MATLAB.

5.1.3 RECONFIGURABLE CARDIAC PROCESSOR

The cardiac processor is designed to process all three cardiac signals concurrently. Each processing chain consists of a feature-extraction block followed by an output-generation block (Fig. 29). In designing the output-generation block, two potential operation scenarios of a sensor node in a future Body-Sensor-Network (BSN), have been considered. In the first scenario, the cardiac sensor node is a distributed node that may control other distributed nodes of the BSN (e.g., nodes delivering BP treatment). In such a case, energy efficiency and data compression is of utmost importance. Accordingly, the first output generation block interprets abnormalities in the raw cardiac data through analog-computation stages, thereby achieving a physiologically-relevant compression of the raw cardiac data in an energy-efficient manner. In the second scenario, the cardiac sensor node is a centralized node that performs real-time

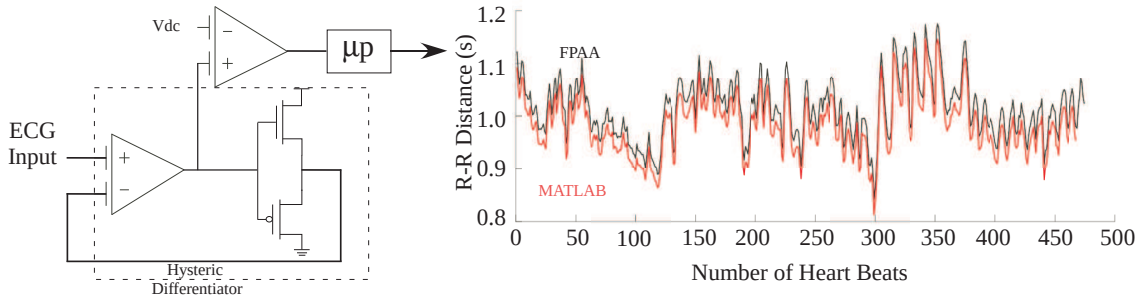


Figure 30: ECG R-wave features are detected by a hysteretic differentiator followed by a voltage comparator. The output of the voltage comparator, which is a train of pulses, is fed to the microprocessor for finding the R-R distance. Consistency is observed in the exemplary waveforms displaying the R-R distance calculated by the on-chip microprocessor using the features detected by the circuit implemented on the FPAA fabric and MATLAB using the features found by MATLAB.

vital-sign streaming to a data-storage unit that the user/clinician can access. Consequently, the second- output generation block has been implemented as a digital algorithm run by the on-chip microcontroller, where the compressed analog outputs are logged to PC in the event of a vital sign abnormality.

The FPAA has been programmed using an open source clone of MATLAB and SIMULINK called SCILAB and XCOS, which provide a graphical interface for the user [29]. The interface allows for rapid prototyping and calibration of the system compared to design an ASIC.

5.1.3.1 Feature Extraction from Physiological Data for Vital Signs

Feature-extraction circuitry is designed that is robust to noise and variation to detect the features described in Subsection 5.1.1. The design leverages FG MOS devices serving as analog memory elements to tune the circuit with patient-dependent parameters.

The R-wave of a heartbeat, which is critical for heart rate calculation, is immediately preceded and followed by points where the magnitude of the first time derivative of the ECG signal is maximum. Therefore, the R-wave detection circuitry is designed

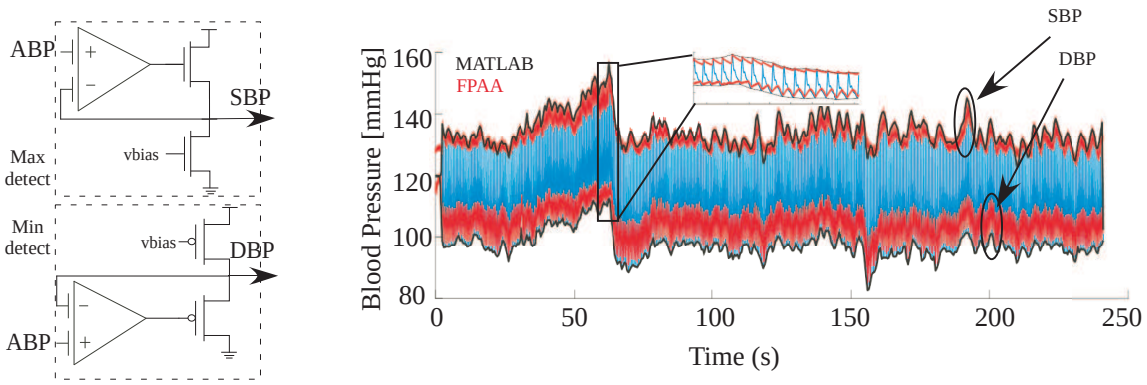


Figure 31: SBP and DBP are detected from the ABP waveform via maximum and minimum detection circuitry, respectively. Exemplary SBP and DBP waveforms (red) obtained from a the ABP (blue) of a healthy subject are shown. The SBP and DBP values obtained by MATLAB (black) are also provided for comparison.

as a differentiator with a time constant set by the transconductance of an FGOTA and a load capacitor, CL1. An hysteric differentiator as shown in (Fig. 30) is used for this computation [43]. An FGOTA comparator converts the differentiator output into clock pulses fed to the microcontroller for calculation of the HR value.

The SBP and DBP values, which are respectively the maximum and minimum values of the continuous BP waveform, are detected by a envelope-tracking circuitry. Fig. 31 shows the output of such a envelope-tracking circuitry.

For monitoring the oxygen saturation level, SpO2, and respiration rate (RR) the ratio, R, of the perfusion indices of the PPG waveforms at red and infrared wavelengths is extracted. The perfusion-index-ratio extraction block consists of circuitry detecting the first order features, namely the peak-to-peak and DC values of red and infrared wavelength PPG signals; followed by arithmetic-operation circuitry for calculating the ratio of the perfusion indices (i.e., peak-to-peak amplitude normalized by the DC value of a PPG waveform) of red and infrared wavelength PPG signals. Peak-to-peak detection is achieved using the envelope detection circuitry described in the BP feature extraction section. The DC values of the PPG signals are detected through a low-pass-filter implemented as a Gm-C stage with a corner frequency set

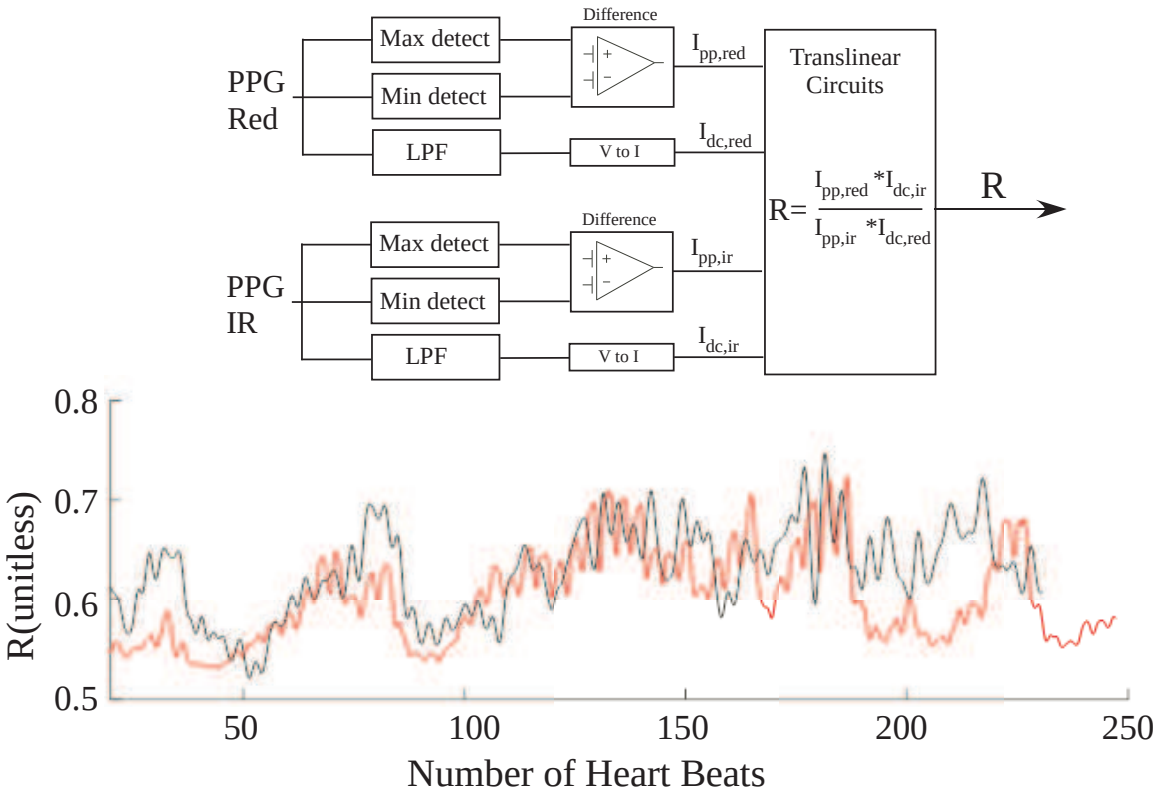


Figure 32: To calculate the peak-to-peak and DC value parameters required to calculate the perfusion indices of red and infrared wavelength PPG signals, max-min detect and GmC low-pass filter stages are used, respectively. The voltage outputs are converted into current using transconductance elements to achieve the perfusion index ratio operation in the current domain using a simple translinear current multiplier/divider. The perfusion-index-ratio waveforms obtained in the analog domain on the FPAA fabric and on MATLAB are smoothed and shown in the plot. They follow similar trends and have similar peaks and valleys. The FPAA output has faster changes owing to values calculated by max, min and DC vary with time.

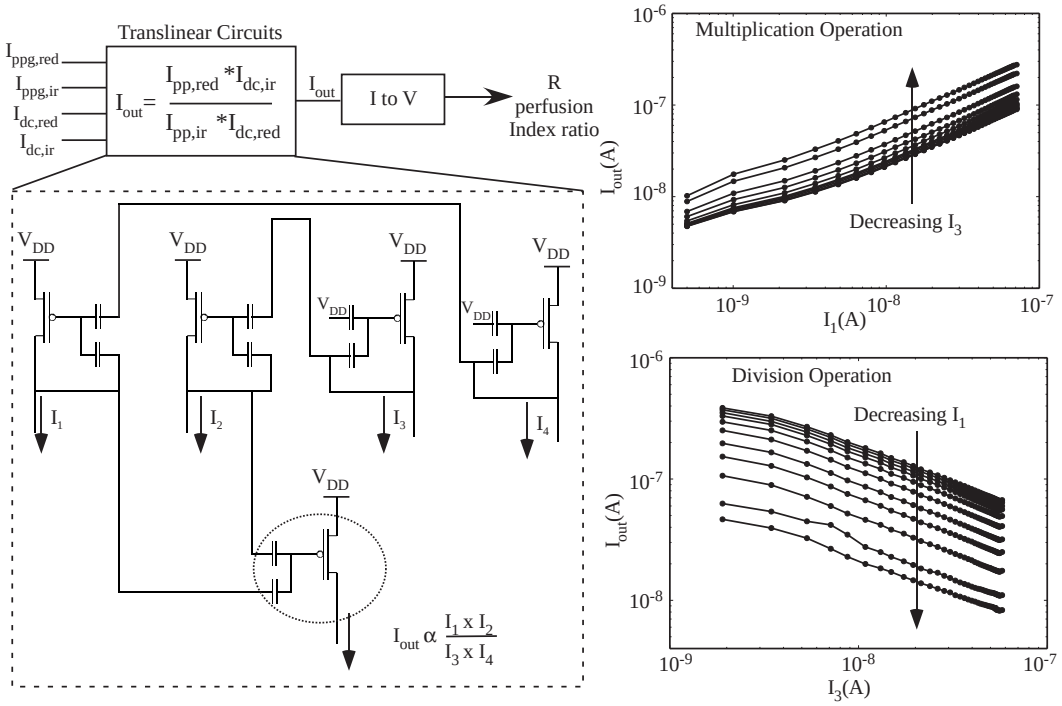


Figure 33: Translinear circuit built using Multiple Input Translinear Element (MITE). Left: Measurement results for translinear circuits for multiplication and division operation. Top: Translinear elements as part of the oxygen monitoring system to calculate the ratio of perfusion indices. The output of the translinear circuit (I_{out}) is converted to voltage using a transimpedance amplifier.

to 300mHz by the transconductance of an FGOTA and a load capacitance, CL3 (Fig. 32). The inputs to the next stage are output currents of wide input-linear-range ($\approx$ 1.5 V) FGOTAs fed by the voltage outputs of the envelope-detection and low-pass-filter circuitry.

For ease of computation and therefore reduced design complexity, the calculation of perfusion indices as well as their ratio is performed in the analog domain using current signals. This is important from the standpoint of a real-time monitoring system where the use of an analog-to-digital converter continuously will substantially increase the power consumption of the system. This also reduces the amount of computation which needs to be done on the processor. The overall calculation is reduced into a single translinear multiplier/divider circuit using multiple input translinear elements (Fig. 33), where the output current, Iout, is dependent on I1 to I4 through

$$I_{out} \propto \frac{I_1 I_2}{I_3 I_4} \tag{16}$$

The relationship in (16) follows from the drain current for FG pMOS in subthreshold regime operated in saturation, and the fact that current is mirrored [96,97]. This is illustrated below starting with drain current in the subthreshold regime:

$$I_{out} = I_{TH} * e^{\frac{\kappa(V_{DD} - wV_1 - wV_2 - V_{TH})}{U_T}} \tag{17}$$

$$I_{out} \propto e^{\frac{-w_1 V_1}{U_T}} e^{\frac{-w_1 V_2}{U_T}}. \tag{18}$$

where w_1 is a product of κ and the FG weight w. V_1 and V_2 can be replaced in terms of I_1, I_3 and I_2, I_4 respectively as follows:

$$V_1 \propto -ln(I_1)\frac{U_T}{w_1} - V_3 \tag{19}$$

$$V_2 \propto -ln(I_2)\frac{U_T}{w_1} - V_4 \tag{20}$$

$$V_3 \propto -ln(I_3)\frac{U_T}{w_1} - V_{dc} \tag{21}$$

$$V_4 \propto -ln(I_4)\frac{U_T}{w_1} - V_{dc}. \tag{22}$$

$$\tag{23}$$

Replacing (5.1.3.1) in (5.1.3.1) will give:

$$I_{out} \propto e^{-\frac{w_1}{U_T}(-\frac{U_T}{w_1}ln(I_1)+\frac{U_T}{w_1}ln(I_3))} e^{-\frac{w_1}{U_T}(-\frac{U_T}{w_1}ln(I_2)+\frac{U_T}{w_1}ln(I_4))} \tag{24}$$

$$I_{out} \propto \frac{I_1 I_2}{I_3 I_4}. \tag{25}$$

The above sets of equations assumes that the weights on the FG are equal and that they are matched. A more general derivation is given in [97].

Iout is converted into voltage, V_R, with a transimpedance stage implemented as an OTA with a feedback transconductance implemented as an FGOTA in a buffer configuration. This voltage output is used for SpO2 critical-level detection circuitry discussed in the subsequent subsection.

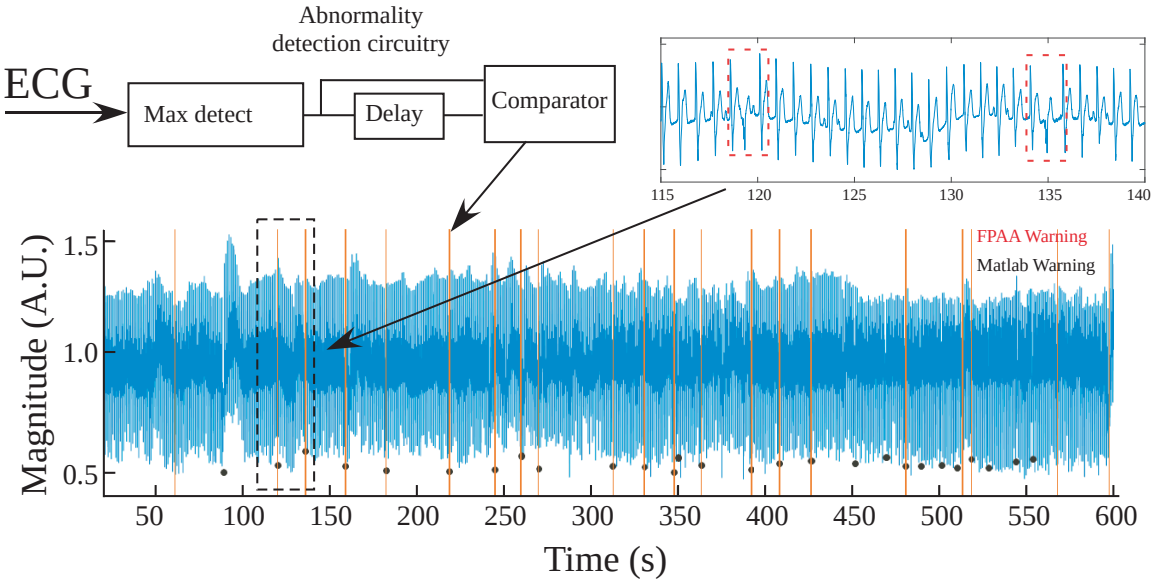


Figure 34: For arrhythmia warning detection, the maxima waveform of an ECG signal tracked by a peak-detection circuit is fed into a comparator. The threshold for the comparator is a delayed version of the ECG maxima waveform. This output is compared with warning generated by MATLAB to analyze the accuracy of such a system. ECG signal of a subject suffering from sinus arrhythmia. The output detects the sudden change in heart rate. The system is tested for 600s of ECG data.

5.1.3.2 Abnormality Detection Circuitry

Abnormalities in the HR (i.e., arrhythmia [35] and SpO2 an SpO2 level below 90% [98]) are both detected in real time and time stamped as pulses. Analog processing is used here successfully to detect sinus arrhythmia from ECG data and critical level of blood oxygen (SpO_2).

Sinus arrhythmia manifests itself as irregular R-R intervals, resulting in prolonged R-R intervals for some heart beats [99]. To detect such an irregularity, the R-R interval is first time-integrated using a peak-detector circuitry (Fig. 34). To a first-order approximation, charge leakage of the load capacitor can be modelled as a constant current source reducing the CL potential by an amount proportional to the R-R interval between successive peaks. The output of the peak-detector is compared with

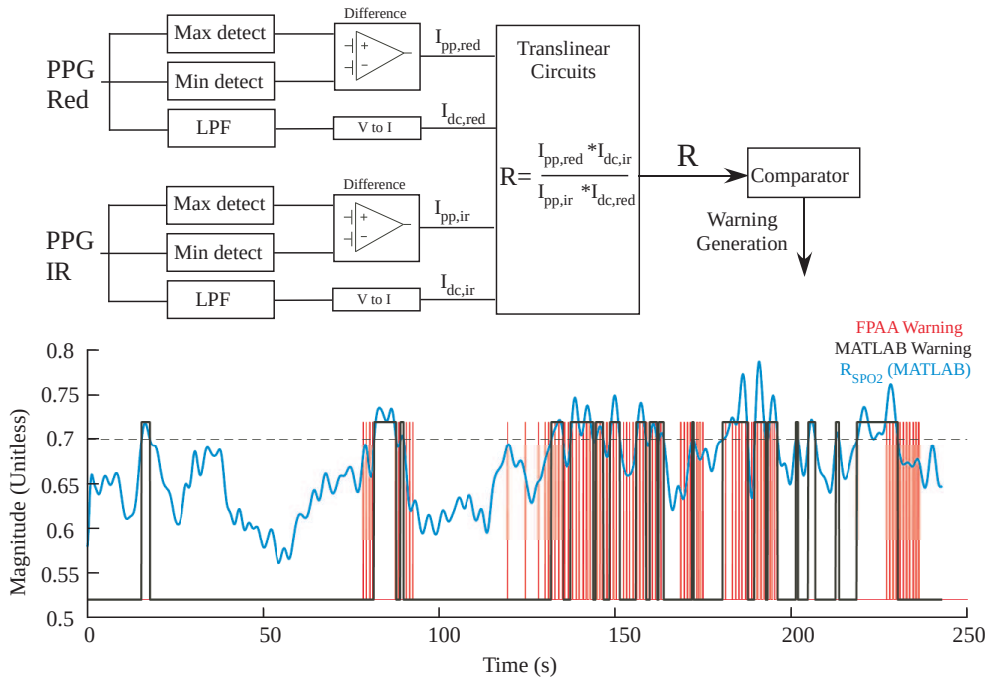


Figure 35: A warning is generated by a comparator when V_R exceeds a preset threshold, implying the SpO2 level dropping below a critical value. The warning generated by FPAA are compared with the ones generated with MATLAB.

a threshold value with a comparator to generate a warning pulse in the event of abnormally long R-R interval. To prevent false positives during normal gradual R-R interval increases (e.g., drops in the HR following deep breathing or exercise), an adaptive-threshold scheme is followed, where the peak-detector output is compared with a one-second-delayed-and-smoothed version of itself. As such, an abrupt increase in the R-R interval are detected as abnormality. The delay network consists of eight cascaded low-pass GmC stages (Fig. 34).

The standard procedure to calculate the SpO2 level from the R value calculated by the PPG feature-extraction circuitry is [100] given by

$$\%SpO_2 = 110 - (25 * R), \tag{26}$$

where R is the ratio of the perfusion indices of the red and infrared wavelength PPG signals. Accordingly, to generate a warning signal at the critical saturation

level of 90%, the analog R signal is compared with a preset threshold value of 0.8 (Fig. 35). It should be noted that, gradual variations in R-R may be considered as normal. Therefore, adaptive thresholding is necessary in generating sinus arrhythmia warning signals. However, a SpO2 level below a certain threshold (e.g., 90%) is often considered as dangerous. Therefore, a fixed threshold is used in the SpO2 warning signal generation circuitry.

5.1.3.3 CARDIAC PROCESSOR VALIDATION AND DISCUSSION

For the ECG, ABP, and PPG datasets of the healthy subjects, the R-R distance, SBP, and SpO2 errors between the results from the cardiac processor and those from MATLAB are summarized in Table II. Comparison results in Table II have been performed in the digital domain using MATLAB after equalizing the sampling rates of the analog processor outputs and the digital data.

Considering the mean R-R distances for all subjects, percentage mean R-R errors for the subjects S1, S2, and S3 are 2.95%, 3.75% and 3.55%, respectively. The mean SBP calculation errors are 1.67% (S1), 1.06% (S2), and 6.27% (S3).

Results for warning generation in the case of arrhythmia and low SpO2 level are presented in Fig. 34 and Fig. 35. Over the course of a ten-minutes ECG dataset from a patient with sinus arrhythmia, with seventeen true-positive, four false-positive, and six false-negative values, positive predictive and sensitivity values of 81% and 74% have been achieved. In Fig. 35, SpO2 warning signals are shown when approximately four minutes of PPG data from a healthy subject is streamed to the cardiac processor. With the aim of warning generation demonstration, the critical-level threshold has been set to 92.5%. The time intervals when the warning pulse frequency increases and those where the low-pass-filtered SpO2 calculated by MATLAB match well (Fig. 35). It should be noted that, following a high-level classification [101], arrhythmia-detection sensitivities potentially higher than the simple

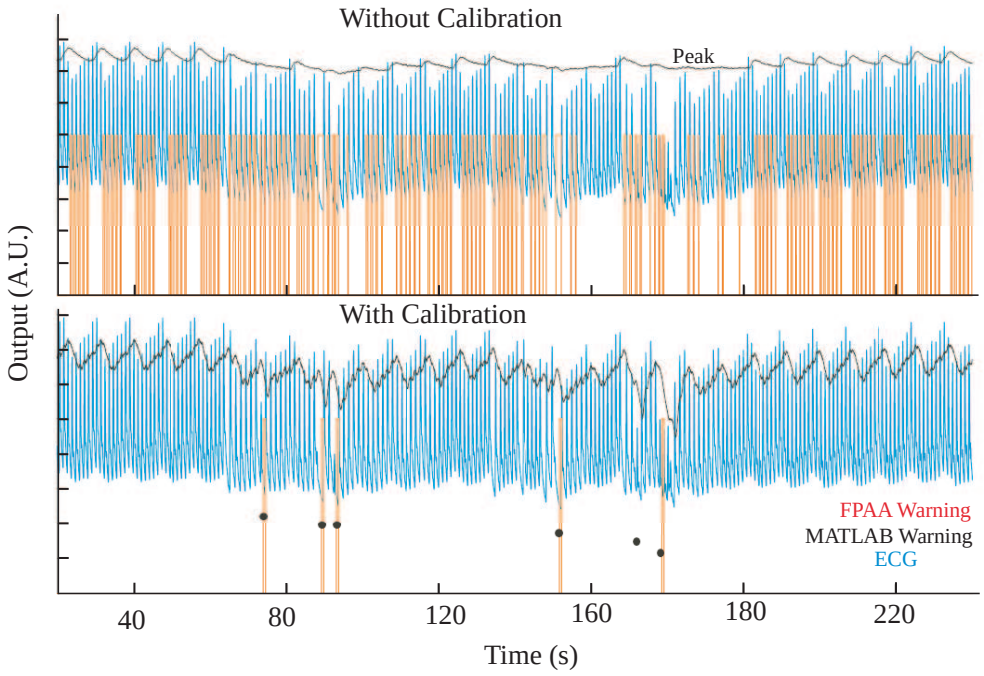


Figure 36: The calibration of the cardiac processor is achieved through FG programming. In arrhythmia detection, a significant improvement is achieved with calibration

threshold-comparison method could be achieved.

Small discrepancies between the MATLAB and FPAA implementations can be attributed to the differences between analog computation and digital computation. Though the algorithm implemented in MATLAB is similar to the one implemented with analog in terms of computation, digital processing is not optimized for real-time implementation since one would require data to be stored before analyzing it. In addition, MATLAB finds local maxima and minima, for the computation, and re-samples the waveform with specific sampling rate. In the case of the analog system, the maxima and minimum are found continuously, and hence provide a better representation of the physiological waveform.

FGs offers huge flexibility in terms of calibrating mixed-signal system and mitigating the intrinsic mismatch present in such a system. From the perspective of a system analyzing physiological and vital signs, this would enable health-care workers

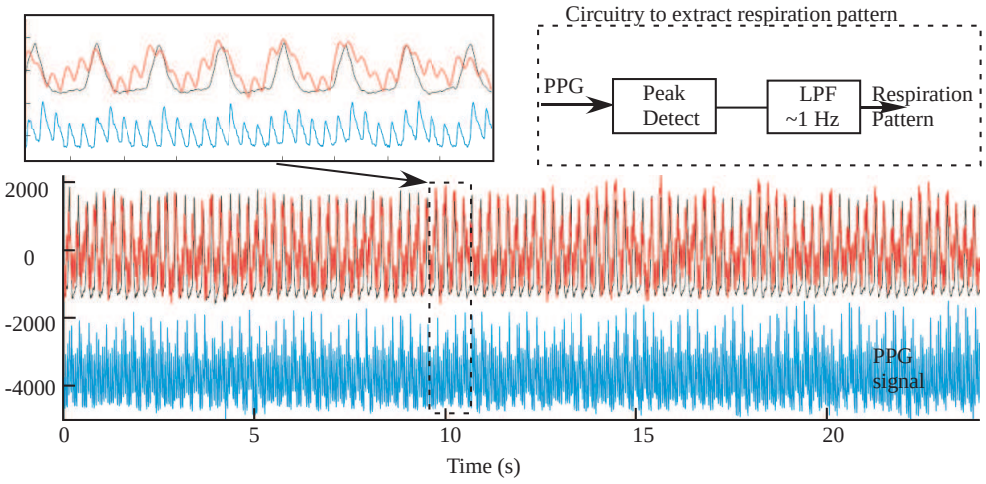


Figure 37: Low-pass-filtered peak detector output from a PPG signal can extract the respiration pattern. The periodicity of the output signal gives the rate of respiration.

to tailor/tune the system to meet the needs of a patient. This process of calibration could be a one-time process at the beginning, similar to the one performed for hearing aids, or could be a continuous adaptation using a built-in self-test system. To illustrate this, we show and compare output of a system before and after calibration. In this case, the system is used to monitor sudden changes in systolic blood pressure, which has a clinical significance in predicting cerebral ischemia. The uncalibrated system shown in Figure 36 has too many false positives compared to the calibrated system and shown in the Fig. 36. As part of the calibration process, one would use a known signal with known targets; in this case, the targets are the warning signal and the processor would program the FG charge of the FGOTA to appropriately change the reference level. In general, one could envision multiple wearable/physiological systems calibrated in similar fashion.

In this study, the focus has been on three vital signs. However, the circuit architectures presented in this work can be used in monitoring other vital signs or cardiac features. For instance, the R-R monitoring approach can be expanded to monitor other periodic vital signs, such as the respiration rate. Changes in intrathoracic pressure during inhalation and exhalation modulates the stroke volume and therefore

Table 6: Power consumption of cardiac processor

System	Power Consumption
ECG R-R detect and warning circuitry	126 nW
SBP and DBP measurement circuitry	251 nW
SPO_2 detection and Warning	1.44 μW
Cardiac Processor Total	1.82 μW

amplitude of PPG signals, which reflects changes in blood volume [102]. Therefore, to monitor the respiration rate, the peak value of one of the PPG signals can be low-pass-filtered (LPF) by a Gm-C filter with a sub-Hz cutoff frequency to obtain the respiration pattern. The similarity between the extracted respiration pattern following that procedure and measured respiration pattern by a pneumatic respiration transducer are displayed in Fig. 37. For the respiration rate calculation, the respiration waveform can be fed to a hysteretic differentiator followed by a comparator for generating pulses at the peaks.

The input dynamic range for the ECG, ABP, and PPG signals is 1 Vpp. For validation of the analog computation modules, signal conditioning has been performed in the digital domain. However, as an ultimate localized processor solution, signal conditioning may be achieved in the analog domain on the FPAA chip [103]. Table 6 shows the power consumption of analog components of the system. Overall, the cardiac processor consumes 1.82 μW with a power breakdown as shown.

5.2 Real-Time Hemodynamic Feature Extraction from Bioimpedance Signals

Electrical properties (e.g., conductivity and permittivity) of biological tissues are unique to the type of tissue (e.g., blood, fat, bone) [104]. Therefore, based on tissue electrical properties, the composition of a tissue (e.g., fat, bone, muscle) and changes in the tissue content (e.g., associated with edema, fluid accumulation) can be monitored. Electrical Bio-Impedance (EBI) measurement is a technique used to obtain these properties of biological tissues by injecting a small-magnitude of AC current into

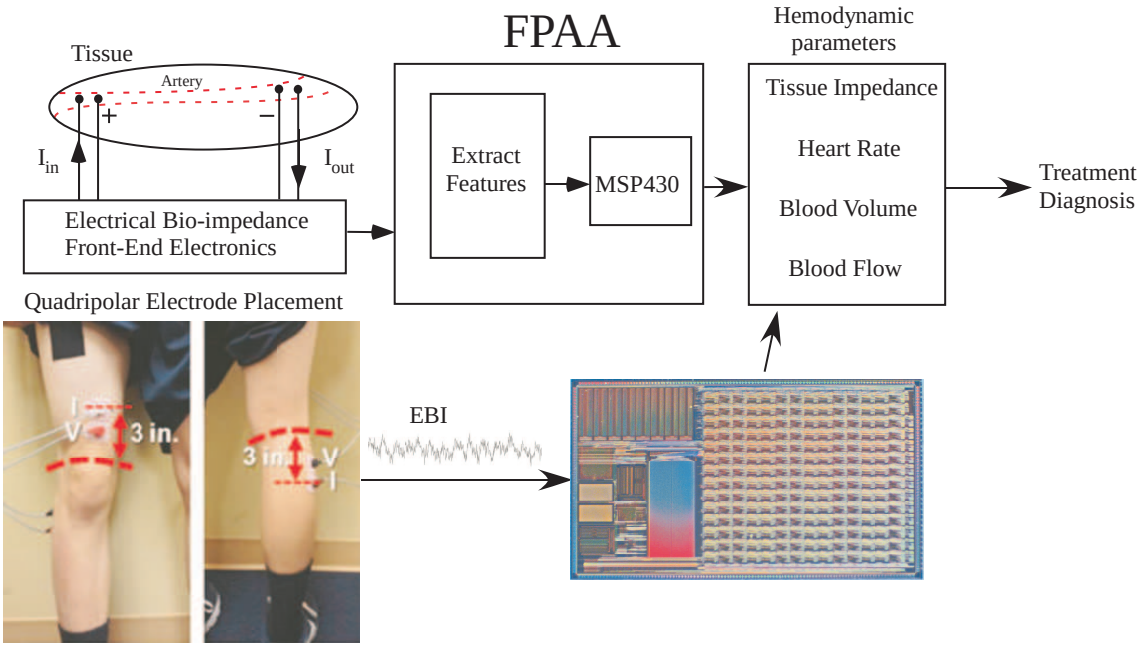


Figure 38: Energy-efficient signal-processing circuitry implemented on an FPAA performs real-time extraction of heart rate (HR) as well as blood volume (BV) and blood flow (BF) variables; from electrical bio-impedance measurements from a knee.

a tissue under investigation and measuring the resulting potential difference across the tissue. It is widely used in research for many applications ranging from extracting tissue composition of a body part (e.g., fat composition [105]) to hemodynamic monitoring (e.g., cardiac output [106]), since EBI is non-invasive in nature and has very high temporal resolution. Additionally, EBI can be measured using low-cost electronics, thereby making EBI a potentially feasible biosignal modality for health assessment applications outside the clinic [107].

Conventionally, EBI systems consist of a front-end block that interfaces with the tissue via electrodes [53] and a digital block for streaming and logging the bio-impedance data to later extract physiologically relevant information (e.g., heart rate, blood flow). While, such a system approach is convenient for systems designed for use in clinical settings, for closed-loop wearable medical applications combining health monitoring and treatment or systems designed for use in resource-limited settings

(e.g., during outdoor activities), an EBI system needs to extract physiological information in real time while not compromising accuracy, energy efficiency, or portability.

Accordingly, this section presents custom-designed, low-power signal-processing circuitry extracting hemodynamics parameters from Impedance Pleythsmography (IPG) signals, which reflect changes in blood volume and are obtained from EBI measurements. To benefit from the power and area efficiency of analog computations while making use of the accuracy of digital, the approach performs computations in the mixed-signal domain, where a FG CMOS-based custom analog-signal-processing circuitry and a digital processor (i.e., processor of a laptop computer) perform the feature extraction and calculations, respectively. For the analog-signal-processing circuitry, an FPAA implementation is preferred over a custom analog design, as the FPAA serves as a flexible silicon fabric where other circuitry corresponding to the remaining components of the ultimate system presented in Fig. 38 would also be implemented. In the ultimate system, the FPAA would also enable tuning the circuit through FG programming to implement subject-dependency (e.g., tuning Iin to ensure high signal-to-noise ratio EBI signals from subjects having different tissue dimensions) and adaptive decision making.

5.2.1 ELECTRICAL BIOIMPEDANCE FRONT-END

Bio-impedance measurements are achieved by injecting a small-magnitude AC current into a tissue and measuring the resulting potential difference. The current is generated via a Wien-Bridge oscillator that generates a 50kHz sinusoidal signal and that is feed into a voltage-controlled current-source delivering the current to the tissue through two electrodes (current electrodes). The frequency of 50 kHz ensures that current could propagate through both intra- and extra-cellular fluid of the tissue. The potential across the tissue is measured via two other electrodes (voltage electrodes) to minimize the effect of skin-electrode impedance and amplified by an

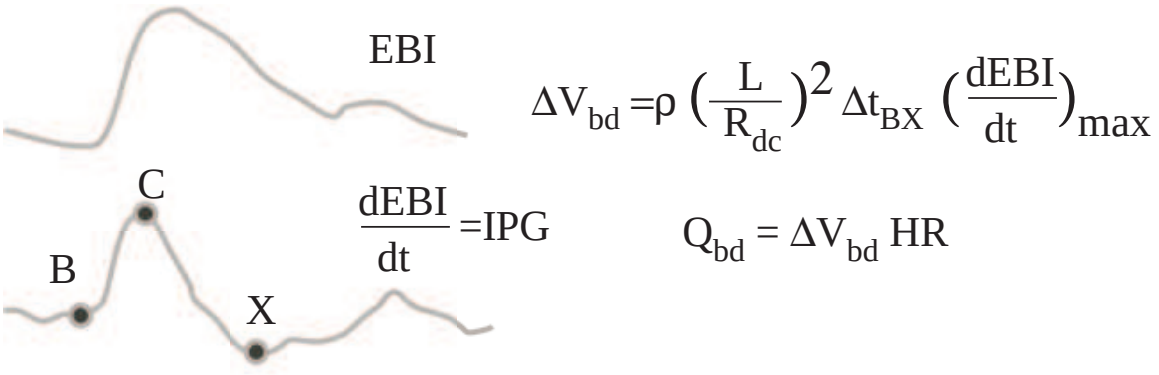


Figure 39: Example EBI and $\frac{dEBI}{dt}$ (IPG) signals (redrawn from [53]) with features important for extracting hemodynamic parameters highlighted. Pulsatile blood volume, ΔV_{bd}, is proportional to the time varying parameters of (i) the time difference between points B and X, Δt_{BX}, and (ii) the peak-to-peak amplitude (i.e., magnitude difference of points C and B) of the rising portion of the IPG signal, namely $\left(\frac{dEBI}{dt}\right)_{\max}$, and the constants of (iii) resistivity of blood (ρ) electrode distance, and the DC resistance of the tissue, R_{dc}. The blood flow rate, namely Q_{bd}, is the product of ΔV_{bd} and the heart rate (HR).

instrumentation amplifier. To obtain the resistance component of the impedance, a phase-sensitive detection circuit is designed using an analog switch controlled by a clock that is in-phase with the injected current. The AC component of the resistance, the IPG signal, which reflects the changes in blood volume, is then obtained using a band-pass filter (BW: 0.1 Hz20 Hz). For more details on the IPG circuitry, the reader is referred to [53].

5.2.2 Analog-Signal-Processing Circuits For Feature Extraction

5.2.2.1 Extracting Hemodynamic Parameters Using IPG

The IPG signal is widely used for hemodynamic monitoring, and can be used to track variations in blood volume as a function of time. Synchronous with the heartbeat, the IPG signal is also used to calculate heart rate. Lastly, from the product of blood volume and heart rate, blood flow is calculated [105] [108].

An example EBI waveform, its first time derivative (i.e., IPG), and the critical

features of IPG described in [109] are presented in Fig. 39. In [108], IPG features are related to pulsatile blood volume (i.e., ΔVbd) and blood flow (i.e., Qbd) using equations presented in Fig. 39. The variables of the equations are described in the Fig. 39 caption.

5.2.2.2 Signal-Processing Circuitry for Feature Extraction

The signal-processing approach taken combines the power efficiency of analog computations with the accuracy of digital. Therefore, the signal-processing circuitry is designed to first identify critical features, namely points B, C, and X, in the IPG, in an energy-efficient manner and then deliver the features to a processor, which accurately calculates heart rate, blood volume, and blood flow.

As a first step, the signal-processing circuitry (Fig. 40) smooths the noisy EBI signal using a low-pass filter with a cutoff frequency of 3 Hz, which is implemented as a first-order Gm-C stage using a 9-transistor OTA. The first time-derivative of the smoothed-EBI signal is obtained using a C4 band-pass filter with a center frequency (3.5 Hz) set to a value greater than the fundamental frequency of the EBI signal (f $\approx$ 1 Hz), thereby differentiating the EBI signal. To compensate for the reduction in gain, the gain is set to 11 dB by increasing the load capacitance, C3. To identify the points B and X, the IPG signal, is fed to a hysteretic differentiator, the output of which changes abruptly when there is a significant change in the IPG [43]. For small changes in the IPG, however, the output almost does not change, thereby inherently suppressing the noisy behavior of the IPG. The hysteretic differentiator output is then converted into a clock, CLK1, which is generated to determine one of the time instances (point B) when the microprocessor digitizes the IPG and to calculate the time difference between the two time instances (points B and X). The signal-processing circuitry generates another clock, CLK2, directly from the output of BPF, to determine the second time instance (point C) when the microprocessor digitizes the

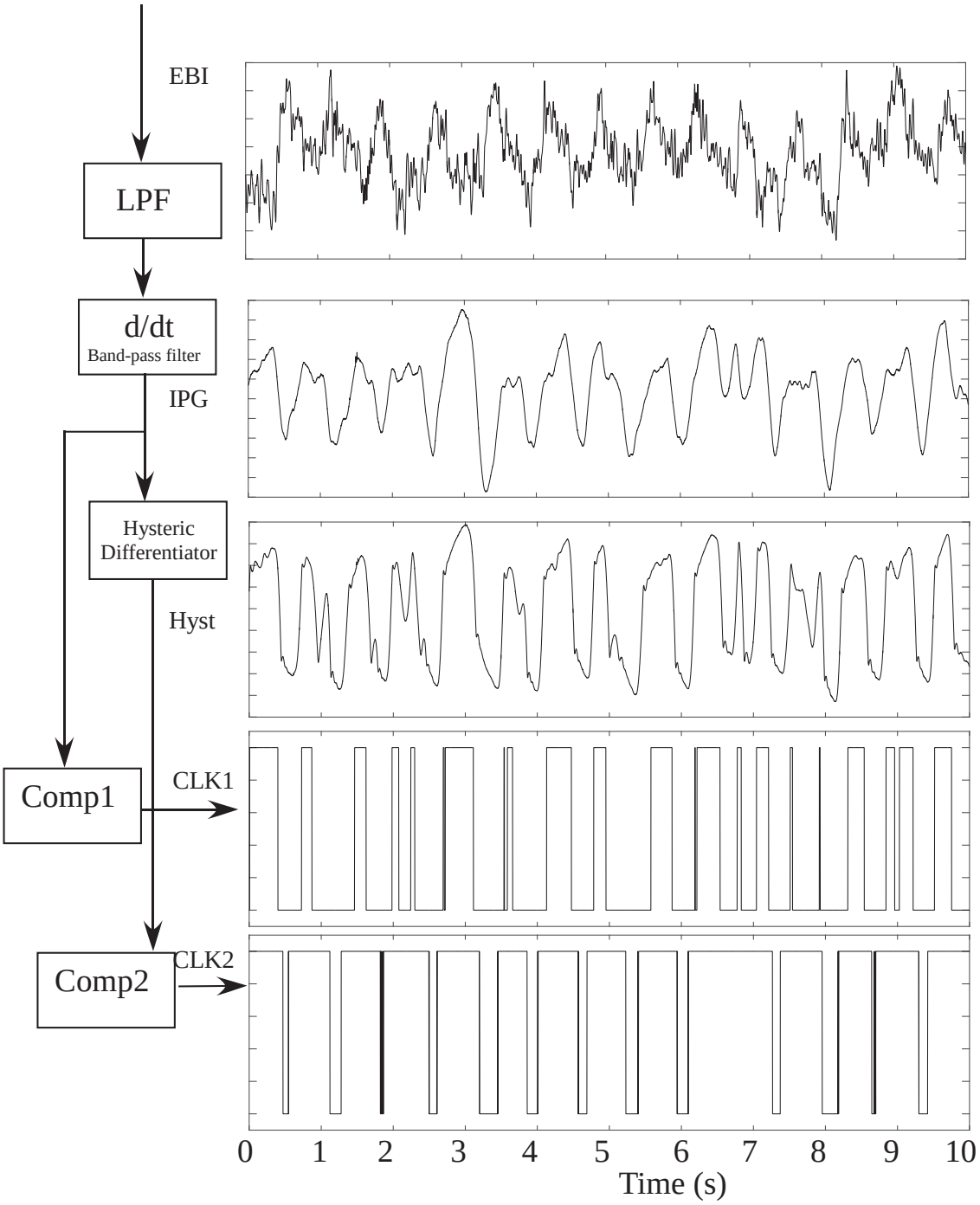


Figure 40: IPG signal-processing circuit. The low-pass filter is used to generate a smooth version of the EBI signal, from which the first derivative is obtained using a C4 band-pass filter. A hysteretic differentiator is implemented to identify the two minimum values of the IPG signal. The figure also shows the input EBI, IPG, hysteretic differentiator output, and CLK1 and CLK2. The waveforms are obtained from 10 s of IPG measurements from the knee.

IPG. Both clocks are generated using comparators implemented as open-loop OTA amplifiers with reference voltages, Vcomp1 and Vcomp2. Example waveforms from the circuit and the resultant clocks are presented in Fig. 40.

5.2.3 Measurement of the system

The analog-signal-processing circuitry implemented on the FPAA has been fed with ≈580 sec. of EBI data, which were collected from the knee of a healthy subject. The data were collected using the analog front-end described in subsection 5.2.1 and a BioNomadix (Biopac Systems Inc., Goleta, CA). The study was approved by the GT Institutional Review Board (IRB) and the Army Human Research Protection Office (AHRPO). Wet-gel electrodes have been used to improve the skin-electrode impedance and data were collected when the subject was seated with legs extended.

For the bias values used, power consumptions are calculated as 0.12 nW, 7.6 nW, 1.25 nW, and 200 nW for the LPF, C4 BPF, hysteretic differentiator, and the comparators, respectively, making a total power consumption of 209 nW. In its current form, the FPAA board is powered from the USB port. In an ultimate wearable design where the FPAA is powered by a lightweight 3.7V, 1000mAh battery, the battery could power the feature-extraction circuitry and the on-board processor (estimated power consumption when the processor is never put into sleep mode: ≈12 mW) for eight days.

For validation of the analog-signal-processing circuitry, MATLAB is used to calculate the heart rate (HR), blood-volume variable (BVV), which is defined as the product of Δt_{BX} and $\left(\frac{dEBI}{dt}\right)_{\max}$ in Fig. 39, and the blood-flow variable (BFV), which is defined as the product of blood-volume variable and heart rate, as well as visualize the results. The high-to-low and low-to-high transitions of the CLK1 correspond to the points B and X, respectively. Point C is where the IPG is at its maximum when the CLK2 is low. For comparison with digital signal processing, a MATLAB

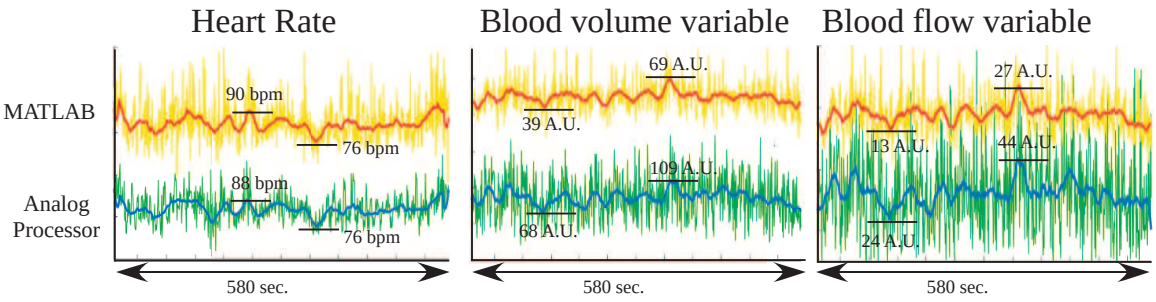


Figure 41: Variation of heart rate, and blood volume and blood flow variables obtained using the analog signal processor presented and the MATLAB implementation display similar trends over the course of a 580 sec. IPG measurement

code performing; (i) time-derivation of the raw IPG signal smoothed using a Savitzky-Golay (SG) Filter of second-order and 201 taps, (ii) identification of the local maxima (points C) and minima (B and X), and (iii) calculation of the HR, BVV, and BFV, is generated. HR, BVV, and BFV calculated using B-C-X features obtained from the FPAA and the MATLAB code are presented in Fig. 41. The smoothed versions of the datasets using a SG filter of second-order and 51 taps show that datasets from the custom signal-processing circuitry and MATLAB display similar trends over the course of the measurement.

5.3 Discussion

The section introduces several circuits and system implemented on a large-scale FPAA SoC. The analog processing techniques would enable devices that could perform continuous physiological monitoring. Data generated from various sensors were processed and analyzed using analog-signal-processing techniques and compared corresponding digital algorithms. The digital algorithms are not optimized for real-time implementation since that would require an embedded platform. DSPs could be used for real-time implementation but that would require the ADCs to operate continuously resulting in significantly more power consumption.

CHAPTER VI

CLASSIFIERS FOR WEARABLE DEVICES

There has been a growing interest in using classification and machine learning for analyzing bio-signals. From a classification point of view, there has been significant progress in the implementation of neural networks and machine learning partly due to the availability of faster devices and partly due to innovation in algorithms. They have been used effectively to solve problems in the field of speech processing, computer vision, and big data. They have also shown promising results in the field of bioengineering and biomedicine to perform medical diagnosis and prognosis [110]. However, most of this work involves performing data acquisition and analyzing it off chips which usually requires large storage and bandwidth for transmission. Such a method would not scale, when taking into account multiple devices generating data for multiple users. Thus a system that could perform efficient and robust computation is necessary where the data can be analyzed and processed locally and in real time.

Figure 42 shows a block diagram of a real-time processing system that should be able to process signals from multiple sensors and classify them into corresponding classes. Eventually, such a system would be deployed for monitoring patients continuously to provide relevant data to the health-care provider. Hence, such a system have to compute and process with only a few μW of power consumption so that they can operate for several days between charging.

From a viewpoint of wearable applications, this chapter introduces a classifier that accurately differentiates between noise and speech in Section 6.1. Further, it shows a real-time processing system that performs a real-time knee-joint health assessment in

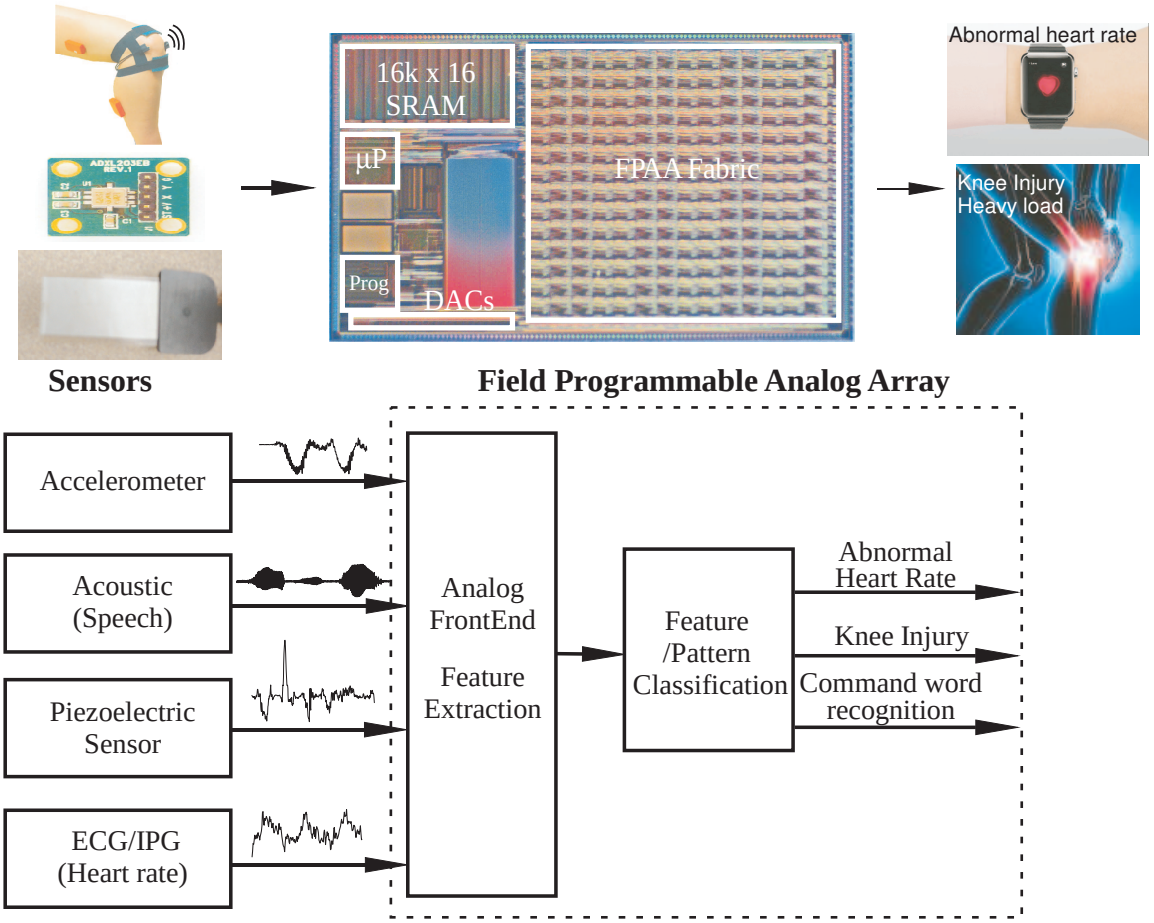


Figure 42: A signal-processing system that takes input from several acoustics and MEMS sensors. For such applications the computational efficiency offered by reconfigurable platform becomes important. The system should be able to process the input and give a relevant output, such as detection of abnormal heart rate or ligament tear based on the classification.

Section 6.2 and Section 6.3. This is important for wearable systems where context-aware computation has become increasingly important.

6.1 Low-Power Speech Detector: Classifying Presence of Speech and Noise

The onset of the internet-of-things has made efficient real-time signal-processing on an embedded platform a necessity. Most digital algorithms require computation in the cloud to which the sensor data must be transferred wirelessly. Thus, in a bandwidth-constrained environment there is a trade-off in the amount of computation done locally

and on a server. Analog computation has been hypothesized to have an efficiency up to 10,000 times more than custom digital processors [111].

Fig. 43 shows a block diagram of a low-power speech detector and its potential application. The system implemented in this work is shown inside the dashed box in Fig. 43. A MEMS microphone, similar to the one shown in [112], can directly be interfaced with the system. As shown in Fig. 43, the system has multiple applications from being used to generate a wake-up signal for a microprocessor for digital signal processing to being used as a startup for analog command word recognition [25] and analog speech classification [37]. Acoustic echo cancellation techniques also involves detecting the near-end speech to halt adaptive filtering [113]. It could also be used as a remote sensory node for continuous speech detection [114].

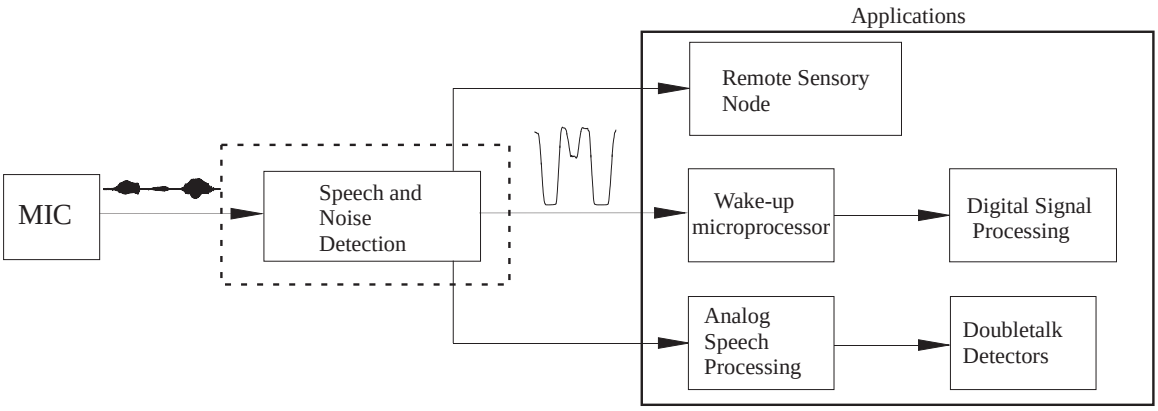


Figure 43: The figure shows an overview of a low-power speech detector and its potential applications. The system that is implemented in this work is shown by the dashed box. The input to the system could be easily interfaced using a MEMS microphone.

6.1.1 Speech Processing using Front-End

Speech processing involves large amount of computation and bandwidth. Thus, a low-power solution for speech processing usually involves a low-power front end that can detect speech over ambient noise and wake the processor when relevant signals are present [115]. Recent efforts in the field of wireless sensors aim at such event-driven

approaches [116].

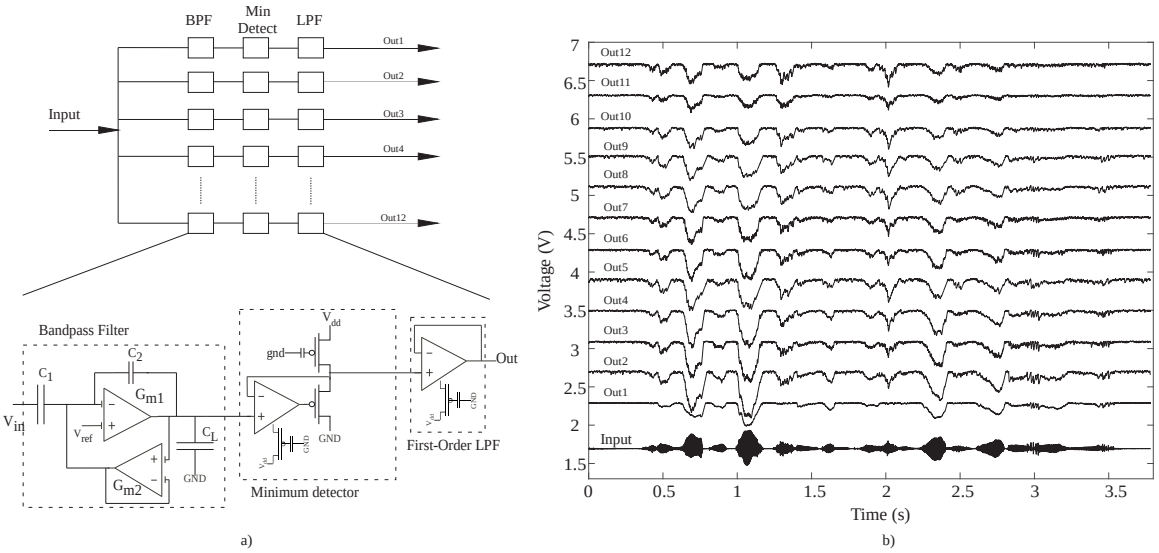


Figure 44: a) A block diagram of the analog front end is shown in the figure along with the circuit schematic used for building these blocks. Input here is from an external DAC, but could be directly interfaced using a MEMS microphone. b) The input and output waveforms are plotted. The signals have been plotted with an offset to visualize them on the same graphs. The analog front end extracts individual features from the speech. These features could be further processed using analog signal processing.

In this work, the low-power front-end is similar to band-pass filters introduced in previous Chapters. The schematic of the band-pass filter is shown in the Fig. 44a. The output of the band-pass filter is passed through a amplitude detect and a Low-Pass Filter (LPF). The schematic of the minimum detector is shown in Fig. 44a. The minimum detector output follows the input when it is decreasing and charges up at a rate of I_{bias}/C_L. I_{bias} is set using a FG transistor biased with a current of 10pA. The LPF further reduces spikes at the output. The LPF is a 9-T OTA configured in a follower configuration with its bias set at 0.9nA, to have a low corner frequency.

Fig. 44b shows the output of these 12 filter banks with its input being a speech signal, taken from the TIMIT dataset. The input from the TIMIT dataset is "She had your dark suit in greasy wash water all year" and the outputs correspond to different

features extracted by the analog front end. An analog shift register controlled by the microprocessor is used to scan the outputs. The outputs have the same DC level and have been plotted in Fig. 44b with an offset. These features are used by the algorithm to learn the weights of the VMM.

6.1.2 Detection using VMM-WTA

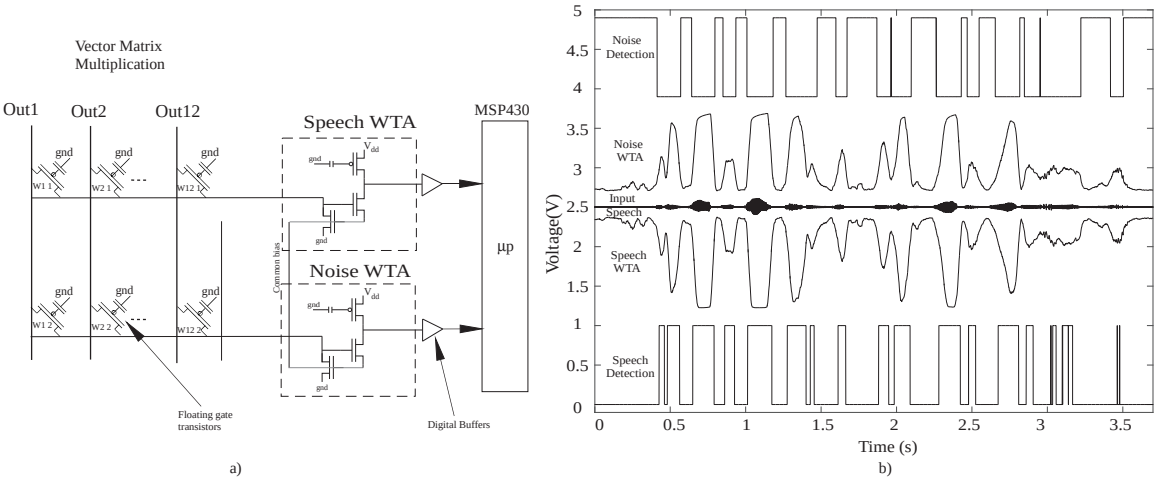


Figure 45: a) Circuit schematic for a 12x2 VMM and WTA are shown here. The output of the WTA are routed to the microprocessor. This signal could eventually be used as an interrupt to the microprocessor for further speech analysis and classification. b) Analog output of the WTAs are plotted along with the digitized output. Here, the digital signal are inverted, to signify a detection when the output is one, as opposed to the WTA output where the winner has a low output.

Fig. 45a shows the transistor-level schematic of a 12x2 VMM. The output of the VMM is a summation of product of 12 inputs and their weights ($I_{out} = \sum WV_{in}$). The weights used for detection are trained off-chip using an algorithm similar to vector quantization. The output of the VMM is fed to the WTA. The schematic of the WTA is shown in Fig. 45a. The WTA is biased such that it allows only one winner at a time. The architecture of the WTA circuit is such that the output is low when it wins and high when it loses. The output of the WTA is routed to the microprocessor using a digital buffer and thus it compares the output with $V_{dd}/2$. The analog output

could also be used as the confidence level of the classification when more than few classes are present.

Fig. 45b shows the output of the WTAs and the corresponding digital outputs are also plotted. The digital outputs are inverted, with respect to the WTA outputs, so as to obtain a digital one when speech or noise is detected. The input given to the system here has a Signal-to-Noise Ratio (SNR) of 20dB. A human auditory system cannot perceive the difference between a speech signal with SNR of 20dB and the one having a higher SNR [117]. The VMM of noise WTA has its weights set such that it has winning output in absence of speech. The speech VMM-WTA is trained to win only when relevant features are observed, i.e in presence of speech. Certain inputs result in a higher level of confidence in detection compared to others.

6.1.3 Accuracy With SNR and Power Consumption

One of the important metrics for real-time speech-processing is to measure the accuracy of the system in noisy environments. To test the system under different noise conditions gaussian white noise was added to the input. The signal was then given as an input to the system using an external arbitrary waveform generator. Fig. 46a shows eight different SNR conditions being tested on the system. Inputs having different SNRs are overlayed with corresponding outputs of the speech detector WTA.

The accuracy of the system is measured not only by checking if the WTA detects the speech but also by measuring the accuracy with which the second WTA detects the noise. Thus, the plot in Fig. 46b considers both as a factor for accuracy. Since the input data is labeled the accuracy was measured by comparing the output with an ideal output. In the case of speech, the error rate was calculated if the WTA is able to detect it or not. For noise, the delay in detection is also considered, so as to reduce false positives. The system performs with an accuracy of 99.94% for inputs having an SNR of 20 dB. The accuracy of the system stays above 70% for SNRs above

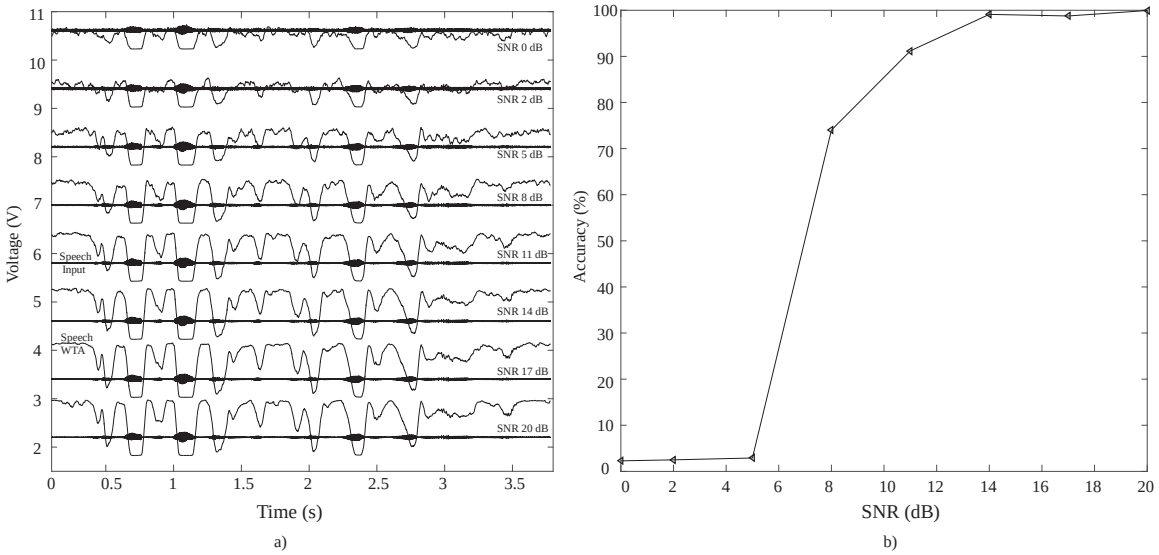


Figure 46: a) Shows the output of the WTA detecting speech for inputs having different SNR. The signals have been plotted with an offset to visualize them on the same graph. b) Accuracy of speech and noise detection with respect to the SNR. White gaussian noise is added to the clean signal from the TIMIT dataset.

7 dB. Below 7 dB of SNR the noise and signal are almost indistinguishable and hence the accuracy falls to almost 2%. The accuracy of the system in low-SNR cases could be improved by having the output of the WTA compared to a different threshold or having several null clusters.

The Speech vs Noise detector has a power consumption of 155.6μW, not taking into account the power consumption of the MSP430. Table 7 summarizes the power consumption for each component of the analog-signal-processing system. The majority of the power is consumed in biasing the G_m-C filters. These can be further optimized by considering the frequency spectrum of the input speech.

Table 7: Power Consumption of Analog Classification System

Components	Power
Band Pass filter(100 Hz - 5 KHz)	150.7μW
Minimum Detector	3μW
Low Pass Filter	27nW
VMM and WTA biasing	1.8μW
Total	155.6μW

6.2 Activity Detector

A promising signal modality to analyze the knee-health condition non-invasively is joint sounds [118] [119]. In previous efforts using joint sounds to probe the status of the knee-joint health, various time- and/or frequency-domain features from acoustical emissions from the knee-joint under motion are extracted and, through sophisticated off-line digital-signal-processing techniques such as single classifiers (e.g., classification using neural networks in [120]) or ensemble of classifiers (e.g., least-squares support-vector-machine fused with dynamic weighting in [121]), variability of the joint-sound features as well as correlations between the features and knee-health status have been investigated. More recently, we have shown that, for a given healthy knee high-frequency acoustical features, namely clicks, of joint sounds consistently occur at similar knee angles for a variety of activities [122]. A distribution pattern of the clicks over the range of knee angle, and an output measure based on the changes in the pattern due to knee disorders could be a promising metric for the knee health. To generate such a metric based on different activities and, therefore, different loading conditions and range of motion of the knee, a wearable system that can operate for hours throughout the day while the subjects perform daily activities is required.

One major challenge towards designing a system that can analyze acoustic sounds from knee joint is the large power consumption associated with high data-rates required (F$\geq$50 kSps) to minimize information loss during the acquisition of knee sounds. In the case of analog systems, this translates into higher bandwidth required for circuits and systems used for analyzing and processing signals. In fact, we have recently shown that a high-sampling-rate data-acquisition system having a wearable form factor can operate only for several hours for recording knee sounds and inertial data for the knee joint. It should be noted, however, that most activities that lead to knee sound emissions last for less than $\approx$15-20 sec. Therefore, by limiting the operation of power-hungry high-bandwidth circuits to operate during

only these valuable activities, as long as the activity detection could be achieved in a power-efficient manner, such a system could be operated for days with limited or no user input. In this Section, we present an early version of a very-low-power real-time activity-classifier that can classify flexion-extension and sit-to-stand activities.

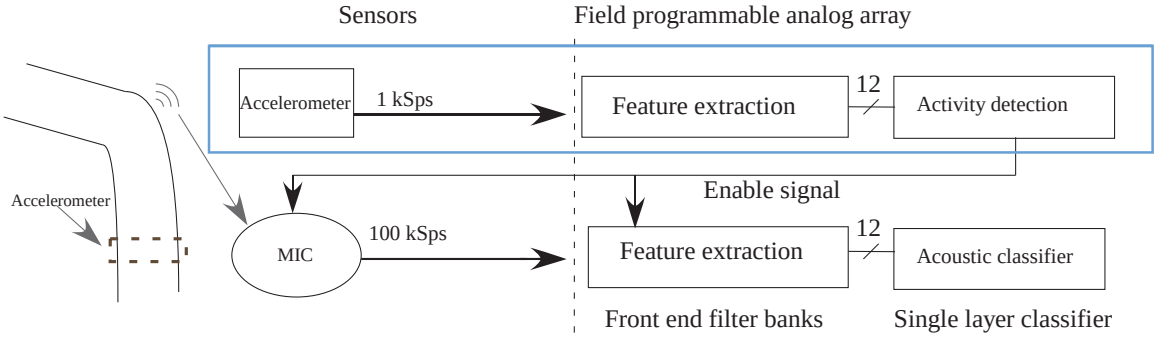


Figure 47: The figure shows an overview of the knee-joint-rehabilitation system. The primary chain implemented in this work is highlighted in blue. The front-end signal-processing chain and the one-layer neural network is implemented to detect the activity in the subject. The activity detector generates an enable signal for MEMS/piezoelectric microphone to start recording.

Fig. 47 shows a block diagram of the proposed system building on the previous work done by the Inan lab [123] [122]. The system consists of a low-power signal-processing chain that extracts the average of the signal spectrum and a classifier using vector matrix multiplication and winner-take-all. Here, a dual axis accelerometer, ADXL203 from Analog Devices, is used for monitoring the activity of the subject. This work uses just one axis (x-axis) of the accelerometer for measurement. The accelerometer has a low sampling rate as opposed to the output of the MEMS and piezoelectric microphone, and, hence the system could operate at lower frequency and lower power compared to the signal chain of the microphone. Thus, the accelerometer signal chain can be used to detect the activity and, in turn, enable the recording of acoustical emission from the knee joint via a microphone.

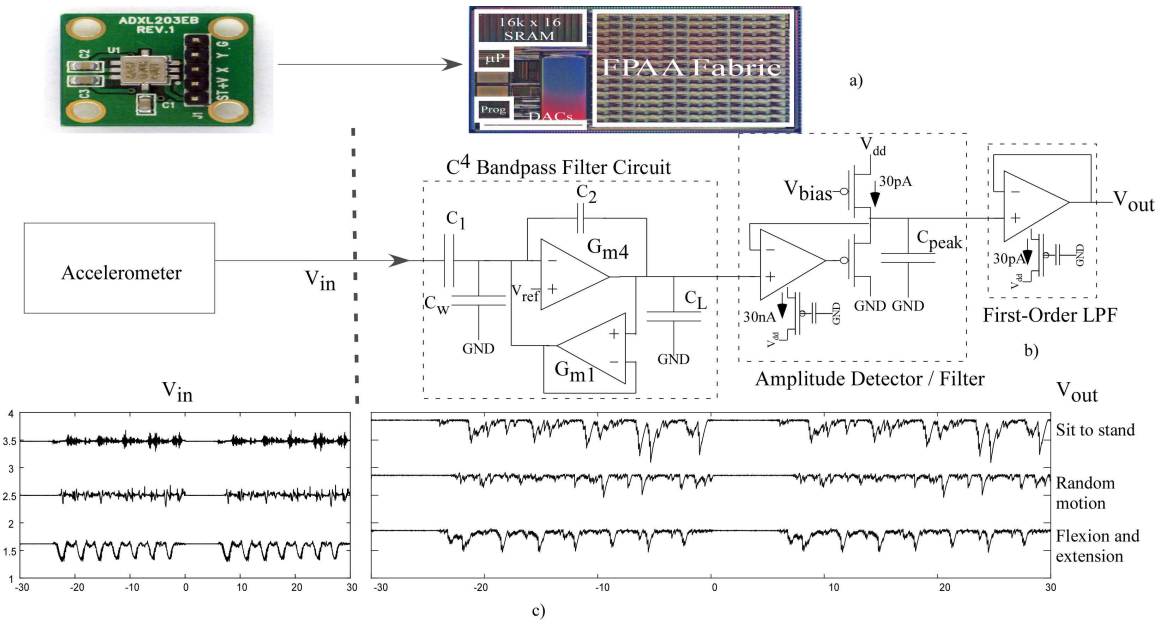


Figure 48: Top of the figure illustrates the interfacing of the accelerometer with a reconfigurable die. b) The figure shows a signal-processing chain to extract the average signal spectrum. The frond end is composed of 12-tap bandpass filters, with a Q of 2 and frequencies spaced evenly from 700 mHz to 15 Hz, a amplitude detect and a low pass filter. c) The output of the accelerometer is recorded for three different activities by the subject. The output response of the fifth channel of the band-pass filter and amplitude detect is shown.

6.2.1 Analog Front-end and Single Layer Classifier

The front end consists of a bank of 12 Gm-C filters followed by amplitude detectors and a low-pass filters. The architecture of the bandpass filter is similar to the one introduced in Chapter 3. The band-pass filter have their frequencies evenly spaced from 700 mHz to 15 Hz with a quality factor of 2. As shown in Fig. 48b, the output of the band-pass filter is passed through a minimum detector and low frequency low-pass filter. The accelerometer is used to monitor different activities of the subject in a controlled environment. Fig. 48c shows the output of the accelerometer during flexion and extension, walking, and sit-to-stand activities. These outputs are then passed through the front end signal processing chain implemented; Fig. 48c shows the

response of the fifth channel, from 12 parallel channels, to these activities. The bandwidth of the low-pass filter could be further reduced to have a smoother response at the output. The power consumption of the front-end processing chain is summarised in Table 8.

Table 8: Power consumption of the compiled front-end analog-processing circuit.

Component	Power (W)
Band Pass (C4)	36 nW
Amplitude Detect	0.3 μW
Low Pass Filter	0.3 μW
Total	0.636 μW

The classifier used for the system is a single layer composed of a VMM and a WTA, described with in detail in Chapter 7. The VMM is used to store weights of the classifier on the floating nodes of FG pFETs. The source terminals of a FG pFETs are used as the inputs of the classifier, since the routing infrastructure has a fixed FG voltage (V_c). The output of a VMM is a current signal which is used by a winner-take-all circuit to make the classification decision. In addition to describing the dynamics and working of both VMM and WTA, Chapter 7 also describes the learning algorithm used for finding appropriate weights for the problem.

Fig. 49 shows the output for the proposed system. A vectorized representation of the signal chain is shown in Fig. 49a. A shift register, controlled using general purpose input/output registers of the MSP430 microprocessor, is used to characterize the signal-processing chain. The root mean square voltages of the 12 filter banks for two different activities are plotted in Fig. 49b.

The weights for the VMM were adapted outside and programmed on the FGs. Here, a 12x2 VMM-WTA is implemented for detecting flexion and extension cycles. Due to the particular topology of the winner-take-all the winning output is active low. For the purpose of testing, a dataset consisting of accelerometer output during flexion and extension cycles and sit-to-stand cycles was created. This dataset was

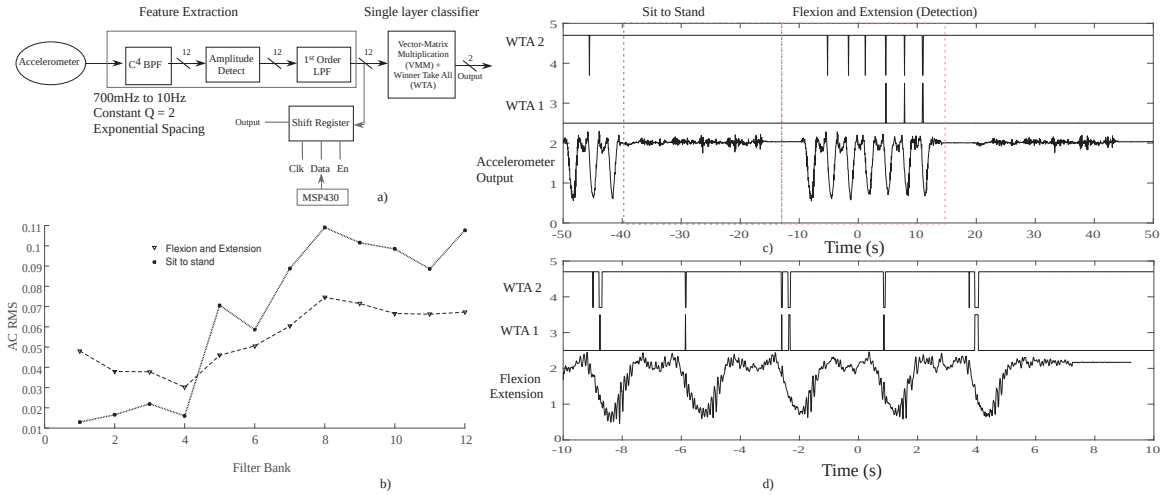


Figure 49: a) Shows the signal chain used to characterize the output of the accelerometer. A shift register is used to scan out the 12 channels of the front end. b)The figure shows AC RMS of two different activities performed by the subject. c) A dataset of two activities, sit-to-stand and flexion detection, was created with goal of detecting flexion and extension cycles in the subject. WTA1 is clustered around a null and wins when the subject is not performing flexion and extension cycles. WTA2 output corresponds to detection of flexion and extension cycles. d) Figure shows the detection of flexion and extension cycles on a shorter time scale.

passed through the signal-processing chain and a single-layer of VMM-WTA. As seen in Fig. 49c, WTA1 has its weights programmed to the values where it wins when there is no flexion and extension cycles and WTA2 is where the detection takes place. Due to finite sampling of the oscilloscope for the time scale used in Fig. 49c, some of the WTA wins are not captured correctly, which can be seen clearly in the experiment performed with the smaller time scale in Fig. 49d. The VMM-WTA structure with its adapted weights, for which the detection takes place, consumes 13 μW of power.

6.3 Classifier for Acoustic Emissions from Knee Joint

This Section will introduce a classifier for assessing knee-joint health. Knee injuries, ranging from simple sprains to ligament tears, are widely prevalent among Americans of all ages. Current techniques for assessing knee joint rehabilitation involve multiple

visits to a clinic. To reduce strain on the health care system and to perform continuous monitoring of the subject, there has been work towards using wearable devices as a non-invasive method for assessing the health information of the knee [92]. It has been shown that acoustical signals can be used to non-invasively measure in-depth information about joint health [57] [55]; wearable devices with embedded acoustical sensors placed around the knee can thus be used as a point-of-care solution for potentially enabling personalized treatment for patients.

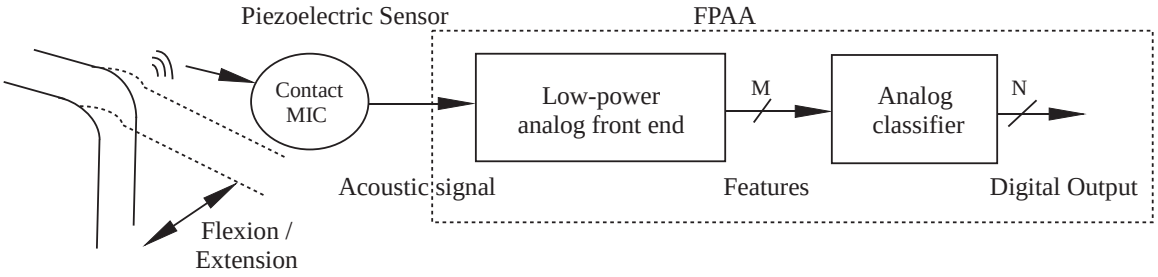


Figure 50: Top level of the proposed acoustic classifier. The output of the analog front-end and the classifier are vectorized. In this work, M is 12 and N is 2.

This Section introduces a proof-of-concept low-power analog classifier for knee-joint health assessment. Proof-of-concept because the system has yet to be tested on multiple subjects. That would require larger dataset or calibrating for each specific patient; FGs would enable that approach. The classifier is used to automatically separate acoustical signatures for a joint with an acute (< 7 days prior) Anterior Cruciate Ligament (ACL) tear compared to the healthy, contralateral side. A piezoelectric sensor (SDT, Measurement Specialties, Hampton, VA) is used as a contact microphone to measure these acoustic emissions. Measurements from a single subject are used as an input for the analog classifier. A top-level block diagram of the proposed system is shown in Fig. 50. The contact microphone recorded an acoustic signal while the subject performed unloaded, seated flexion / extension of the knee. These signals are analyzed by the compiled system comprising 12 parallel second-order band pass filter, amplitude detectors, LPFs, and a single-layer classifier implemented using

VMM-WTA. A system similar to the one presented in the previous Section and in Chapter 3. The WTA used in this work has a digital output and could be easily stored by the processor.

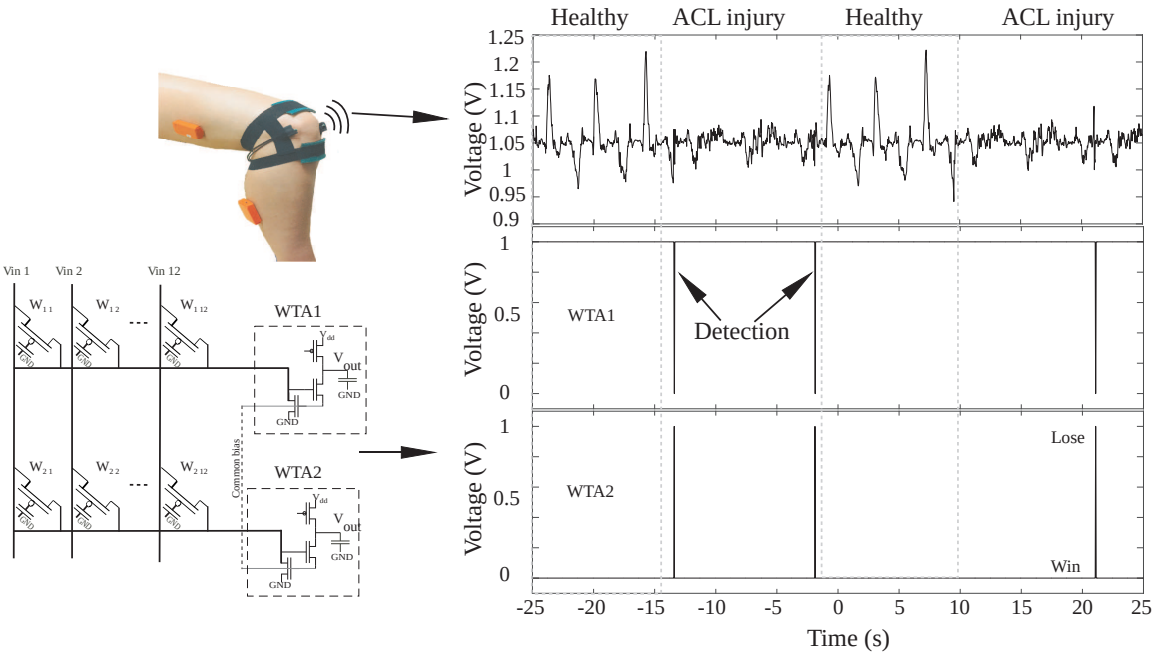


Figure 51: Output of the VMM-WTA along with recording and sensor placement. Two 1-WTA are used for the classification of ACL injury. The data were recorded from a single subject having an ACL injury on one knee. WTA1 predicts the injury by winning when an injured acoustic signal is presented to it. WTA2 wins when the signal is from a healthy knee joint.

6.3.1 Classification of ACL injuries

A single-layer VMM-WTA is used to perform non-linear classification. Since, a single-layer of VMM-WTA can perform XOR classification, it can be considered a universal approximator [70] [124]. Though, for this to be true it might require additional number of nulls to create the right classification boundary. Additional investigation of the system and learning techniques and algorithm needs to be undertaken. Because of this fact, a single-layer of 12x2 VMM-WTA is used to design a classifier for detecting the presence of ACL injury prior to reconstructive surgery. Fig. 51 shows

the recording system and sensor placement during subject testing. Recording from a single subject with an injured and a healthy knee is analyzed and used as input to the system. Data recorded from the same subject are used here to avoid the need for calibration. For the system to be generalized, two things would be essential in making the hardware perform self-calibration for a user and learn weights using a larger dataset.

Acoustic recording of a healthy and an injured knee is passed through the analog front-end, and their outputs are used for training the weights of the VMM. The weights are trained ofline for ease of implementation and faster convergence of the gradient, though this would be the method used the in case of a larger dataset too. Inputs to the VMM are observed using a 16-bit shift register on the chip. The training of the VMM is done off-chip by clustering the weights around their inputs. The FG PFET transistors in the VMM are biased in subthreshold to reduce the power consumption. A schematic of the 12x2 VMM-WTA is shown in Fig. 51. A common bias for the WTAs is generated using a FG PFET and an NFET current mirror.

A dataset comprising of the injured and the healthy knees was created for the purpose of testing the classifier. Fig. 51 shows part of the dataset being used for testing the classifier. WTA1 detects when it is presented with features from an injured knee, whereas WTA2 wins for the rest of the input. The classifier consumes 12.2μW of power. The buffers used at the output of the WTA, to drive I/O pads, consumes 10μW of power and so dominates the power consumption of the classifier. The buffers are used here to route the output to be able to drive the Input-Output pads on the PCB.

6.4 Command-Word Recognition

Another essential part of wearable system is robust command-word recognition. The need for command-word recognition has increased in recent years with the rise of

home assistant devices. Even though cloud based learning system offer command-word recognition they have two major drawbacks power and privacy.

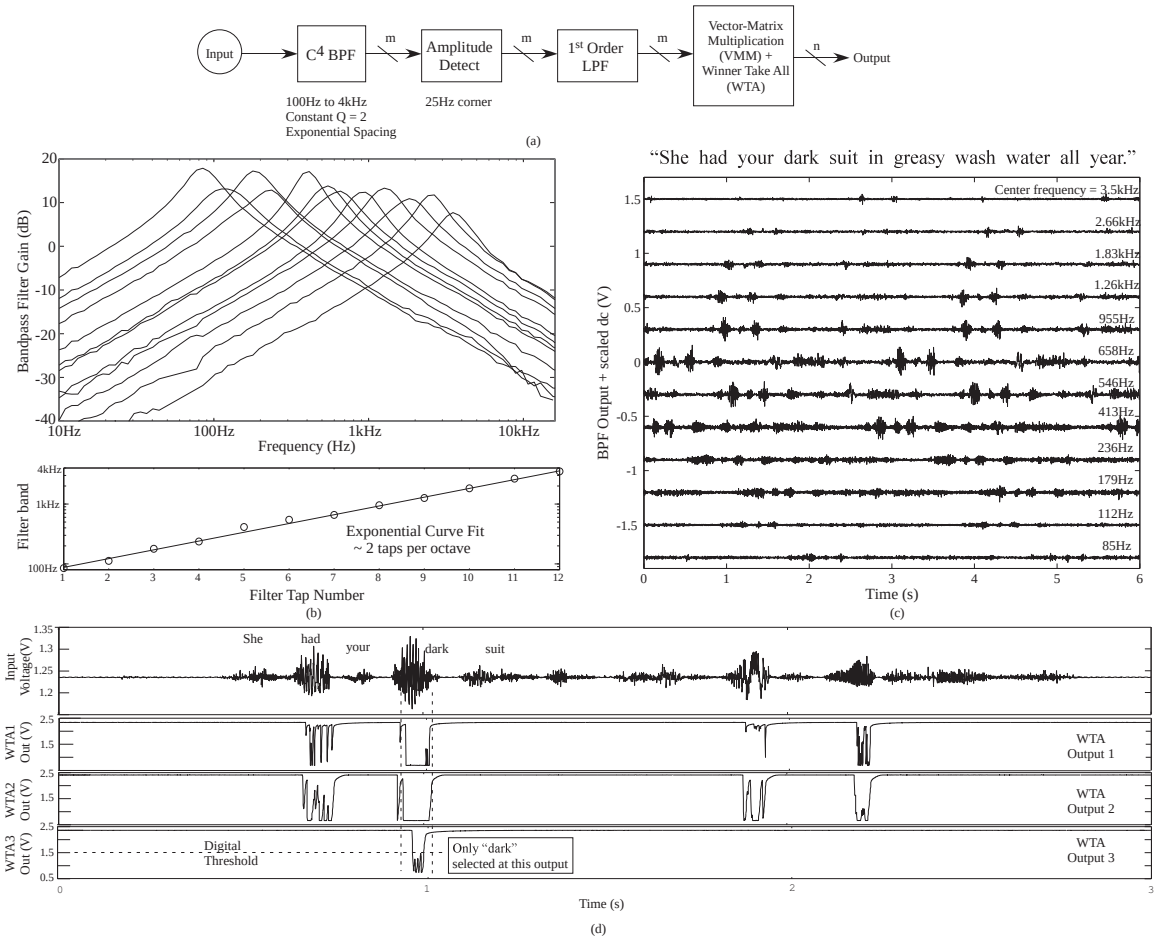


Figure 52: Analog auditory word classification application compiled into the RASP 3.0, showing the experimental waveforms from the IC.

We show an example application of auditory / speech classification looking at detecting a command word in a sentence. Figure 52 shows the first application example of an auditory classifier structure for a limited phrase, like a command word, that can be classified through features in the averaged signal spectrum. Continuous-time spectrum decomposition used a bank of constant Q filters at the first processing stage, using a bank of amplitude detection and filtering operations, and then next a VMM + WTA classifier block to classify each of the resulting spectrum into simple symbols.

In a more complex speech recognition system, one might have the spectrum correspond to phonemes or part of phonemes and build up the temporal representations using temporal classification (i.e. HMM classification) to word spot the resulting phonemes, syllables, and words. A simple command word application requires only to distinguish between a few simple symbols, can be directly computed as a state machine on the MSP430 processor; a next level of computation, using a simple Viterbi decoder, could be directly implemented on the MSP430 processor as well.

6.5 Conclusions

The analog classifiers introduced in this chapter can be used for variety of tasks. Particulary, their use in wearable devices is highly attractive considering energy efficiency is of great importance in such systems. More complex classification tasks can be achieved by biasing the WTA circuit to achieve K-Winners as opposed to one winner and treating the outputs as probability or confidence of classification. Similar classifier systems can also be used for classifying images or for facial recognition. Further, one can envision classifiers with multiple layers for performing more complex classification tasks.

CHAPTER VII

LEARNING FOR VMM+ WTA EMBEDDED CLASSIFIERS

This Chapter focuses on learning for and hardware implementation of embedded classifiers using Vector-Matrix Multipliers (VMMs) and Winner-Take-Alls (WTAs). Having introduced classifiers and their application for various datasets, in particular this Chapter describes in detail the architecture of the system, the learning algorithm and the circuit techniques used to mitigate mismatch in such analog system.

A single-layer of VMM and WTA is a universal approximator [124]. To demonstrate this, it was used to perform the XoR operation [70], which was proved by Minsky and Papert to be a task that a one-layer of perceptron was not able to perform [125]. This enabled using VMM-WTA as a feasible circuit architecture for performing classification for non-linearly-separable tasks.

7.1 VMM+WTA Circuit Structure, Biasing and Mismatch

This Section describes the circuit structure of a VMM-WTA classifier. Further, it discusses the biasing of the WTA and mismatch compensation in a VMM-WTA which arises due to indirect programming of the FG [39].

Fig. 53 shows the architecture of a VMM-WTA classifier on a reconfigurable fabric. The weights are stored as charge on floating nodes of the FG pFETs and are implemented in the local routing of a CAB, as seen in Fig. 53. WTAs are built by interconnecting two nFETs in the CAB [30]. This adds the capacitance of local routing [25]; it would be better to have a WTA as part of CAB elements to reduce parasitic capacitance and leakage current. The circuit derives from Lazzaro's WTA

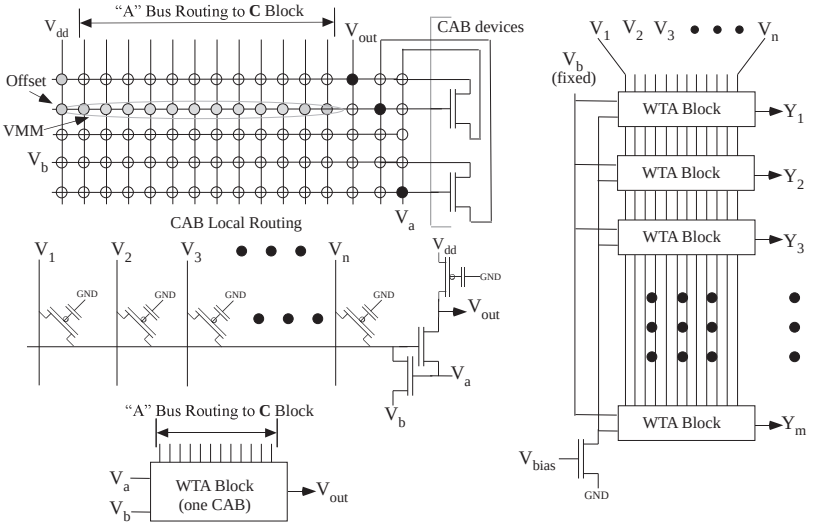


Figure 53: Physical FPAA implementation of the VMM + WTA module in the FPAA. The VMM and the offset are implemented as a row of FG switches connected to the input of the two nFET transistor (current conveyer) configuration. The reduced routing, circuit, and block representation are all shown. This block, implemented in a single CAB (with its two nFET transistors), is replicated in multiple CABs, one CAB each output.

circuit but uses FG pFET devices to enable programmable load devices [71].

Fig. 54 shows possible sources of mismatch in a VMM-WTA classifier. There are several source of variations in such a classifier structure, assuming that the DC level of the inputs to the classifier are perfectly matched. The FG as explained in previous chapters uses indirect programming which suffers from threshold voltage mismatch between the programming transistor and the transistor used for operation. This is denoted in Fig. 54 as ΔV_{To}. The FGs in the VMM and in the WTA suffer from this kind of mismatch and have to be compensated before being used in a classifier. The threshold voltage mismatch (ΔV_{To}) is calculated by measuring the ratio of drain currents flowing through the two FG transistors when operated in the subthreshold saturation regime. The Following equation illustrates the process of calculating the threshold voltage mismatch:

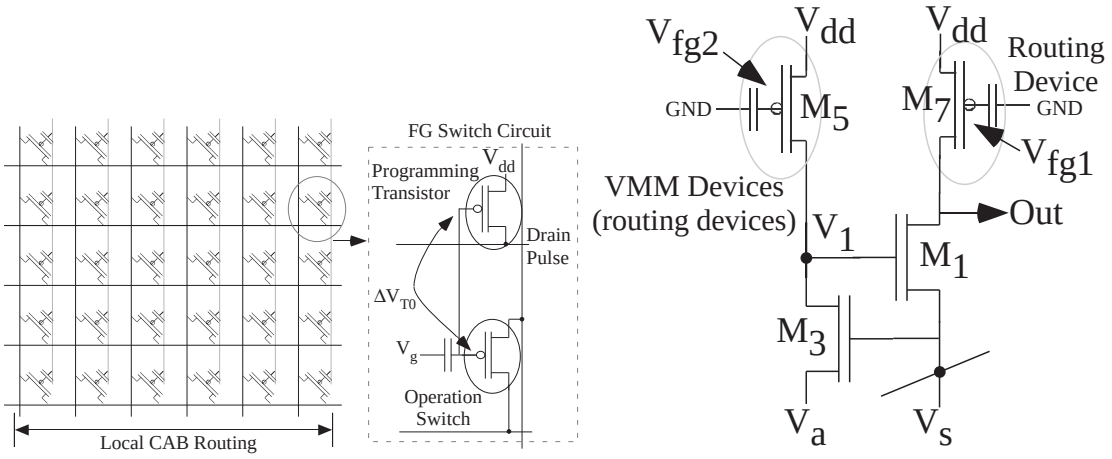


Figure 54: Mismatch in a VMM-WTA classifier structure. To mitigate mismatch one needs to understand the sources of mismatch. In case of indirect programming there is a threshold voltage mismatch between the programming transistor and the transistor used for operation. In a WTA the load transistor, shown in the figure as M1, has a threshold voltage mismatch ΔV_{To} compared to its programming transistor. The FG in the VMM, shown as M5, also suffers from threshold voltage mismatch (ΔV_{To}). The nMOS transistors in WTA also suffer from device mismatch and could potentially affect the gain $\frac{I_{out}}{I_1}$.

$$I_{d1} = I_{TH} e^{\frac{\kappa(V_{DD}-V_{fg1}-V_{To})}{U_T}} \quad (27)$$

$$I_{d2} = I_{TH} e^{\frac{\kappa(V_{DD}-V_{fg2}-V_{To})}{U_T}} \quad (28)$$

$$\frac{I_{d1}}{I_{d2}} = e^{\frac{\kappa(V_{fg2}-V_{fg1})}{U_T}} \quad (29)$$

$$V_{fg2} - V_{fg1} = \Delta V_{To} = \frac{\kappa}{U_T} ln(\frac{I_{d1}}{I_{d2}}). \quad (30)$$

where (27) and (28) are drain currents of FG pMOS transistor in the subthreshold saturation regime. This threshold voltage mismatch can be used while programming the indirect transistor in the programming phase. One can create a mismatch map of all the FG transistors during the calibration phase of the FPAA.

The above mentioned technique of creating mismatch map mitigates the mismatch that arises from the indirect programming method. Another source of mismatch is the nMOS transistors in the WTA, shown as M_1 and M_3. This results in varying gain between different WTA blocks. Assuming the nMOS transistors are operating

in subthreshold saturation, it follows:

$$
\begin{aligned}
I_1 &= I_{TH1} e^{\frac{\kappa(V_S - V_{To1}) + \sigma V_1}{U_T}} \\
I_{out} &= I_{TH3} e^{\frac{\kappa(V_1 - V_{To3}) - V_S + \sigma V_{out}}{U_T}} \\
\frac{I_{out}}{I_1} &= \frac{I_{TH1}}{I_{TH3}} e^{\frac{\kappa((V_S - V_1) + (V_{TO3} - V_{TO1})) + \sigma(V_1 - V_{out}) + V_S}{U_T}} \qquad (31) \\
\frac{I_{out}}{I_1} &= \frac{I_{TH1}}{I_{TH3}} e^{\frac{\kappa(V_{TO3} - V_{TO1})}{U_T}} e^{\frac{\kappa(V_S - V_1) + \sigma(V_1 - V_{out}) + V_S}{U_T}} \qquad (32)
\end{aligned}
$$

In (31) the notation is similar to the Fig. 54. The first two terms in (32) arise from the mismatch between the two nMOS transistors. One could also write in term of change in input current to output current.

$$
\frac{\Delta I_{out}}{\Delta I_1} = \frac{\kappa}{\sigma} \frac{I_{TH1}}{I_{TH3}} e^{\frac{\kappa(V_{TO3} - V_{TO1})}{U_T}} e^{\frac{\kappa(V_1 - V_S) - \sigma V_1 - V_S}{U_T}} \qquad (33)
$$

One way to reduce the effect of this is by characterizing these transistor a priori and compensate by adding the mismatch while programming the VMM and the load FG pMOS of the WTA. Another approach involves using the concepts of BIST where the system self calibrates itself. Here, we partly follow the direction of BIST where known currents are programmed into the structure and from that measure the limits, in terms of difference in current it can discern, set due to the mismatch and other factors.

Figure 55 shows the measured operation of the WTA circuit embedded in a VMM and WTA learning classifier structure. The weight matrix (12x8) is programmed to an identity matrix illustrating the operation of each WTA input / output stage. This identity matrix is programmed (5nA) on top of a 10nA baseline current. This measurement uses on-chip DACs to enable each input (2.4V to 2.5V), in turn, to enable a single current for each WTA input. Given the input pattern, we expect the outputs to win, in sequence, from the first output through the eighth output,

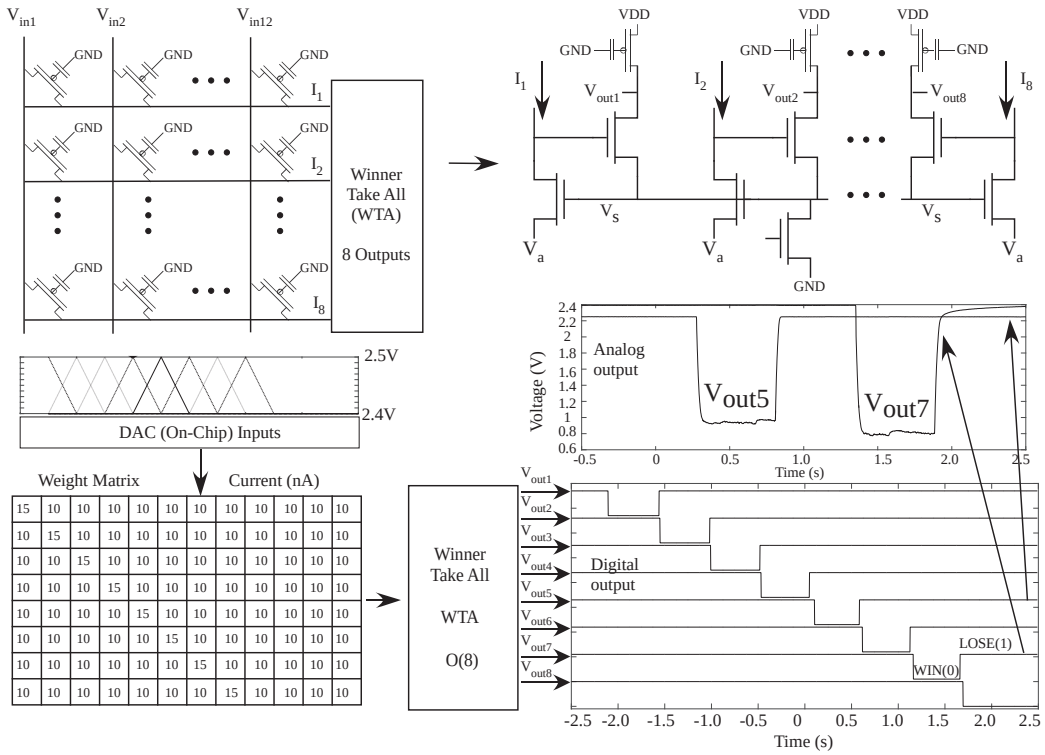


Figure 55: Illustration of VMM-WTA circuit and minimum current required to be programmed in the VMM for WTA to make a decision with confidence. The weights on the VMM are programmed to identical currents (10nA). The diagonal elements of the matrix are programmed to 15nA. Eight DACs are used as inputs to the VMM. Inputs to the DAC are such that they output voltage to create a dominant current in each column sequentially. Here 2.5 V is used as 1 and 2.4 as 0 since FG pMOS exhibit exponential behaviour with respect to source terminal in subthreshold region of operation. The eight WTA outputs all each win in sequence. The particular measured output waveform moves between a losing signal (between 2.2 V and 2.5 V) and a winning signal (below 1.2V). The winning signal is limited by the voltage of the common bias (V_s) on the WTA line.

corresponding to the experimental measurements. The load devices are programmed such that the WTAs are biased in a region where there can only be one winner.

The experiment performed in the Figure 55 has two objectives:

- Experimentally find the minimum current required for WTA to make a decision with confidence.
- Biasing of WTA, load FG pFET and nFET bias, is such that it only allows one winner.

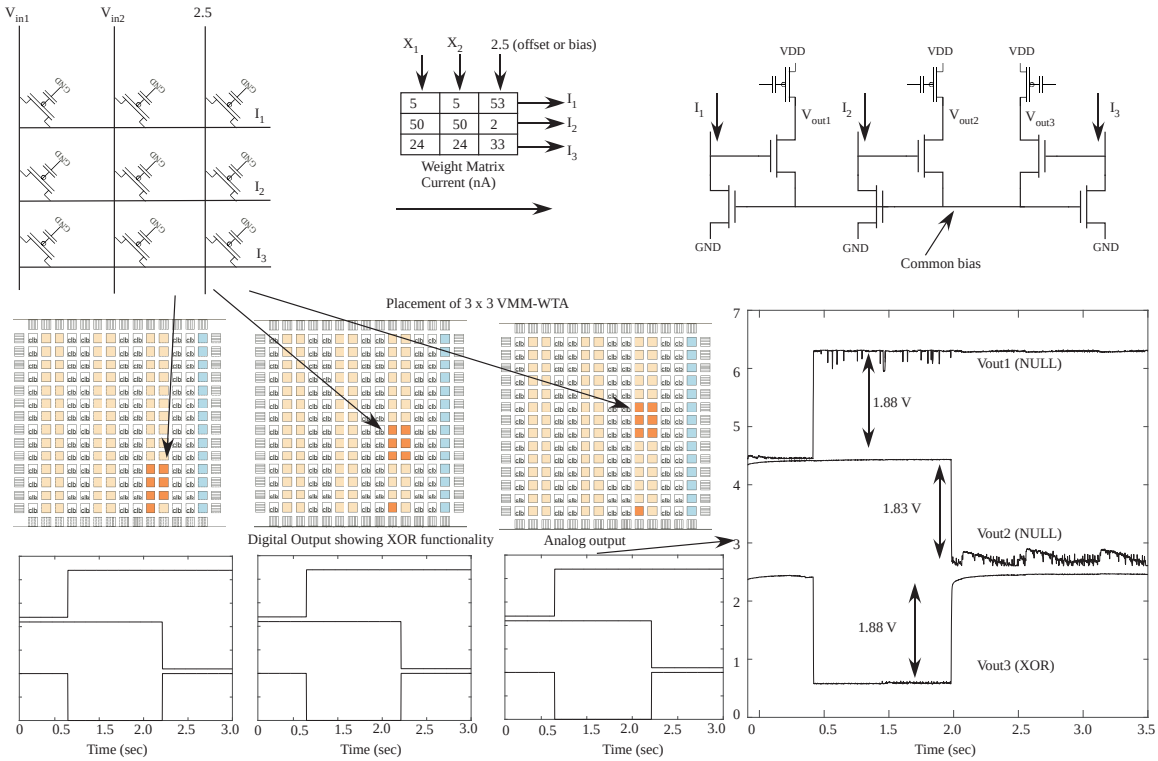


Figure 56: The XOR classifier data is repeated for the VMM at three different locations, as seen by the three VPR routing views, and similar results. Multiple locations show the calibration eliminates effects of ΔV_{T0} due to indirect programming. The location of the VMM weight matrix has little effect on the resulting computation due to initial measurements that calibrate the V_{T0} mismatch from indirect programming.

These tasks are important from an analog classifier perspective where mismatch and variations could change the classification boundary substantially. In Fig. 55, the WTA is able to make a decision by discerning a difference of 5nA. The output of the WTA is between 2.2 V to 2.5 V if it looses and is between 0.9 V to 1.1 V if it wins. The Fig. 55 demonstrates that the mismatch map created was successful in compensating for the mismatch.

Another test before designing a classifier is to check the robustness of the VMM-WTA structure when different weights are programmed. In other word even though the structure is shown to be a universal approximator [70] is it robust in presence of the mismatch. In case of this mismatch is not just the variability in threshold voltage

of the FG pFET but also the difference in input DC level. As an example the 7-bit DACs used in the measurement of Identity matrix shown in 55 can have different analog values for 0 and 1. From the measurement performed on a chip two DACs which were given the same hex value, digital input, had difference of few 100mV. This difference has a large changes (exponential decay) in the current of the FG pFET since they are source driven. Depending on the precision of the initial calibration [42] the effect of this variability has to be analyzed and tested. There is also variation in internal V_{DD} between different CABs, hence during the creation of mismatch map the current measured to calculate the threshold voltage difference used internal V_{DD}. To check the robustness of classifier structure an XOR classifier was compiled on to the FPAA as shown in the Fig. 56. The XOR structure is compiled over different CABs to verify the calibration and mismatch map process eliminates the variation discussed earlier. Figure 56 also shows the locations where the XOR classifier has been compiled. The analog output of the XOR classifier has a difference of almost 1.9 V between the winning and the losing WTA.

7.2 Classification of Acoustic data

Chapter 6 introduced several different classifiers. In this section an acoustic dataset created for Nzero program [116] is used for classification. The dataset is composed of acoustic emissions from truck, generator, and car with human speech in the background. The goal of such a classifier would be to distinguish between various auditory sounds and act as a context-aware device for cloud based classifier. The goal is not to create the most precise classifier but to design one which could reduce the power consumption of a wearable device/remote node significantly. It could also be a self-powered or battery-less system and in that scenario energy efficiency takes a higher precedence over the classification accuracy. A better figure of merit would be power consumption over correctly classified inputs.

7.2.1 Algorithm for learning

The algorithm was developed through understanding the connections of VMM+WTA classifier to Self Organizing Maps (SOM) [126], Vector Quantization (VQ) and Learning VQ [127], and Support Vector Machine (SVM) capabilities [128]. Training multilayer networks, typically required for universal approximation, requires propagating the error backwards to learn the weights [129]. This usually requires large amount of data and computation power. Hence, typically a learning task is usually performed on a GPU to attain convergence in a feasible amount of time.

The algorithms developed for machine learning are optimized for digital processors and hence when implementing on an analog/mixed-signal platform requires significant modification [130, 131]. The classifier learning algorithm uses a combination of clustering and least-mean square gradient descent to learn the weights of the classifier. The following equation demonstrates the clustering step of the algorithm:

$$w = (yhat * x')/norm((yhat * x'))' \tag{34}$$

where yhat is the target for the learning algorithm, x is the input to the classifier. In case of this classifier x would be the output of the filter bank structure. The clustering step involves also updating null outputs. For example the XOR example uses two nulls and one output. Subsequent steps in the learning involves adapting the weights depending on the error between the target (yhat) and the actual output of the classifier ($y = w'x$). The last column of x is the constant offset (bias) 2.5 volts (digital 1). Since least mean square is a good error metric we use the following step to adapt the weights:

$$w_{new} = w_{old} + (((yhat - yout) * x')./norm((yhat - yout) * x'))' \tag{35}$$

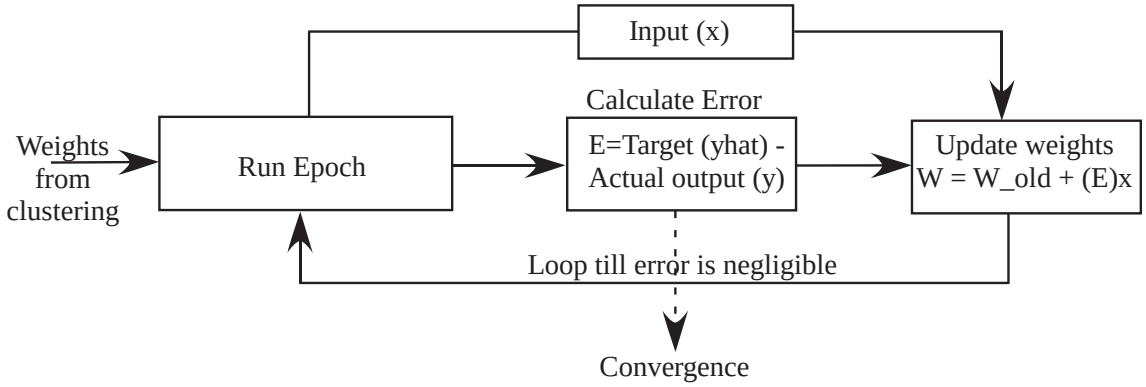


Figure 57: Weight adaptation algorithm implemented. Weights are adapted till the error is negligible. The learning is performed in batch.

The adaptation in 7.2.1 is performed till the error ($error = norm(yhat - yout)$) is small. The output stage is a winner-take-all circuit biased to have only one winner. Hence, depending on the input to the WTA ($y = w'x$) its output would either win (yout=1) or loose (yout=0). The hardware implementation of the WTA is inverse that is when the WTA wins the output is 0 and when it looses it is 1.

7.2.2 Twelve-input Classifier Learning Experimental Measurement

Figure 58 shows a comparative experimental measurement results for on-chip classifier and one where learning and classification is performed off the FPAA for comparison. The input dataset utilized a larger dataset composed of measured background acoustic sounds and additional measurements of generators, idle cars, and trucks in this environment. The input dataset was then composed of multiple 1s bursts of a generator, idle car, or truck on a 110s background; constructing the dataset in this way produces a labeled dataset. All learning and classification occurred on this dataset passing through the same frequency decomposition stage: 12 overlapping bandpass (C^4) filters, 12 peak detectors, and 12 low-pass filters (LPF, 50Hz corner for spectrum representation). FGota is used to level-shift the output of the front-end close the power-supply rail. This stage is important since the FG transistors in the VMM are driven from there source terminal and are programmed using a source voltage of

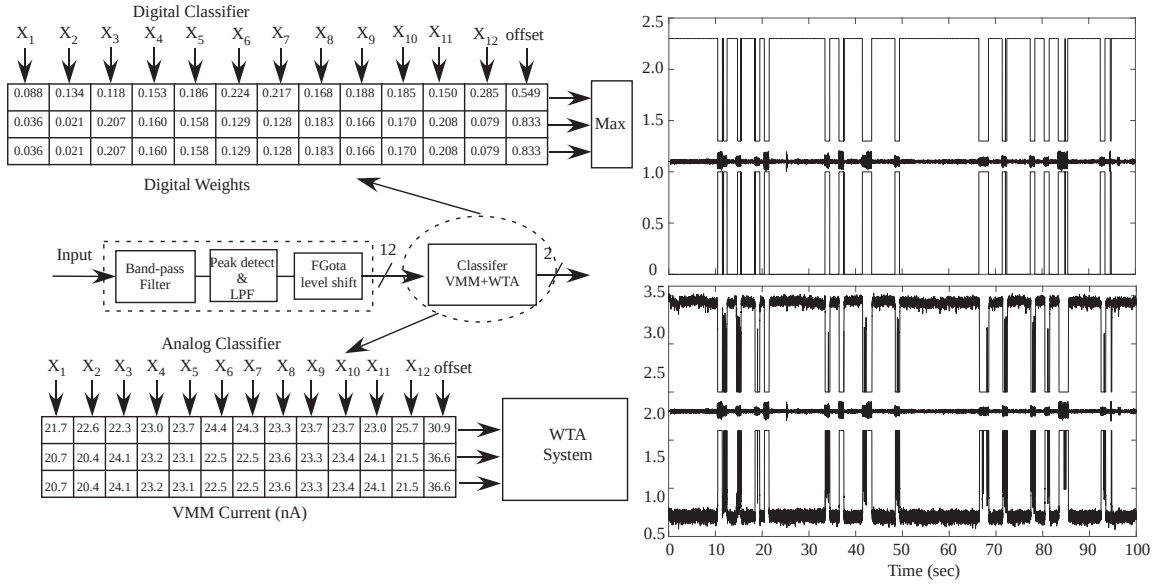


Figure 58: VMM+WTA classification of an acoustic dataset created using a series of 1s data inputs, identifying the presence of a sound source, whether it be a generator, truck, or car. The classifier used a 12x3 VMM classifier followed by a 3 input, 3 output WTA. Both on-chip classification and off-chip classification using circuit models are shown for comparison. In both cases the output of the analog front-end measured on-chip are used. The data measurements were offset to show the input signal, the WTA output (top vector), and one WTA null (third vector) on the same plot. These two approaches yield similar results, and assuming a minimum time for any symbol of 40ms, the classifier correctly recognized the results every time.

2.5 volts. Hence, any shift from this would result in an exponential decay of current.

The approach in 58 shows a sensor-to-classified signal processing chain, unlike most classification algorithms, including hardware based classification and training algorithms. The power consumption of this classifier structure is $24\mu W$, including the front-end analog circuits. Due to this analog classifier are well poised for battery-less or self-powered systems. In general, operating these classifiers as a context-aware wake up devices could achieve substantial power reduction in wearable systems.

7.3 Discussion

7.3.1 Computation required for VMM + WTA learning classifier

The equivalent digital computation of this classifier, between the bandpass filter operation as well as the equivalent 12x3 VMM operating at a slow rate of 200SPS is roughly 4MMAC (/s). A Multiply ACcumulate (MAC) unit operating near the energy efficiency wall [17] will take roughly 1mW of power, consistent with the similar processing and energy requirements of digital hearing aid devices.

The resulting memory access is likely a factor of 2 to 8 larger than this computation [17]. Implementation on an embedded processor, would require 250pJ/Op, typical of low power processors, would require roughly 4-8mW for these numeric computations. A typical ADC for this computation, such as ADI7457 [132], would require 1mW at 3.3V supply to transform the resulting acoustic signal to the digital processor. The required classifier levels (23μW) are significantly less than the required, dedicated digital computation.

7.3.2 Size of Classifier Implementations on SoC FPAA device

The section describes the maximum size of a neural network that can be compiled on a single SoC FPAA device. A network could be built as a single layer network, or as a combination of layers; each VMM+WTA classifier is a universal approximator for its input / output space. The maximum problem size depends on the number of WTA stages and then on the number of synapses and inputs. One can get between 1-2 WTA stages per CAB, with 98 CABs on the IC. The current implementation uses 16 inputs per CAB, although this number can be increased significantly by using the C block switches in addition to the local routing switches. Configurable fabric can allow for sparse patterns, which could potentially improve the computational complexity as in digital systems; in this case, we look at fully connected local arrays to provide one possible metric on this design.

CHAPTER VIII

CONCLUSION

The subject of this research is implementation of real-time signal processing on an embedded platform. As an embedded platform the work uses a floating-gate based field programmable analog array. The work also demonstrates embedded classifiers for various wearable applications and physiological monitoring. Several techniques and systems which help in reducing mismatch and variation that are observed in mixed-signal circuit have been demonstrated.

8.1 Research Summary

Chapter 2 described the basic structure and programming of Floating Gates (FG) and there use in analog signal processing. The chapter also discusses the evolution of FG Field Programmable Analog Array (FPAA). It introduced the state-of-the-art FG based FPAA which is predominantly used in this work.

Chapter 3 described a Built-in Self Test (BIST) system to tune analog front-end used extensively for feature extraction in analog classifiers. Specifically, the capacitively coupled current conveyer circuit which performs band-pass filtering of the input signal. In general, it could be used to tune multiple parameters on a chip.

Chapter 4 described techniques for reducing temperature variability of several different circuits on a reconfigurable platform. It also introduces models for simulating temperature behaviour of various analog circuits. Measurement from several current and voltage reference are presented in the Chapter.

Chapter 5 showed the application of reconfigurable analog platform for monitoring

vital signs. Various physiological signals are analyzed using low-power analog processing techniques. In particular four different physiological signal namely electrocardiography, blood pressure, photoplethysmography and impedance pleythsmography are analyzed.

Chapter 6 introduced several different analog classifiers. It discusses a classifier which distinguishes between speech and noise signal. The chapter also showed measurement results from a system which is used as an activity detector analyzing signal from an accelerometer. Also, a proof-of-concept classifier analyzing acoustic signals from the knee-joint to determine the presence of ACL injury.

Chapter 7 described in detail the different aspects of implementing an analog classifier on the chip. It introduced calibration of the circuits and systems used as part of the analog classifier. The learning algorithm used for training the classifier is also described. The Chapter also showed measurement results from a VMM+WTA classification of an acoustic dataset created using a series of 1s data inputs, identifying the presence of a sound source, whether it be a generator, truck, or car.

8.2 List of Contributions

- **Built-In-Self-Test (BIST):** I performed the design and measurement of BIST system introduced in chapter 3. In this work the front-end of a classifier, which is used to extract features from the input signal, is used as a design under test. It becomes critical in a large classifier to accurately tune the several different parameters accurately. A journal paper [133] was published in TCAS-I.

- **Rasp**3.0**:** Testing and development of tools for the current FPAA. This involved helping with development of the scilab library, VPR architecture files, assembly program for executing commands on MSP430 processor and helping with the development of the remote FPAA server . This is a collaborative work with current (Sihwan Kim, Aishwarya Natarajan) and former members of ICE

lab. This resulted in following publications [25] [134] [42] [30] [135].

- **Rasp**3.1**:** Design and fabrication of next generation of FPAA in collaboration with Sihwan Kim and Aishwarya Natarajan. Measurement and testing of these ICs have not been performed.

- **Temperature Robust Systems On A Reconfigurable Platform:** The work on temperature robust circuits and systems allows for modeling and designing circuits and systems which are robust over variation in temperature. The modeling part of the work modified the EKV models developed by Aishwarya Natarajan published in [60]. Designing of the voltage reference and the system were done in collaboration with Hakan Toreyin. I performed the measurements of the circuits and system over different temperature range using the Cincinnati Sub-Zero: Environmental Test Chamber. The work has been submitted to TVLSI journal and is under review [136, 137]

- **Speech vs Noise Detection:** The work on detecting speech in presence of noise is critical for various low-power speech classification systems. The accuracy of the classification was measured for various SNR levels. I performed the designing and measurement of the system. This work has been accepted in Circuit and System, ISCAS 2017 ,conference [56].

- **Command Word recognition:** The work used TIMIT dataset to recognize the word "dark". This was published as an application of the FPAA [25]. In this work I designed and measured the command word recognition system.

- **Low power activity detector** The use of FPAA as a context aware signal processing unit is demonstrated by using it for detecting Flexion and Extension cycles and'Sit to Stand activities performed by a patient. This work was in collaboration with Dr Omer Inan's Lab and the accelerometer data was taken

by Hakan Toreyin. I used the accelerometer data to classify it on the FPAA. The work was published in EMBC conference [55].

- **Classification of Knee-Joint Sound** Acoustic data from a healthy and a injured knee was given as an input to an on-chip classifier. The work was in collaboration with Dr. Inan's lab and the acoustic data was measured by Caitlin Teague. Using the data from the piezoelectric sensor I created a dataset for training the classifier weight off-chip. The weights were then compiled onto the FPAA for classification. This work was published in IEEE Sensors Conference [65].

- **Real-Time Hemodynamic Feature Extraction from Bioimpedance Signals** This work presents custom designed and low power signal processing circuitry extracting hemodynamics parameters from IPG signals, which reflect changes in blood volume and are obtained from EBI measurements [54]. The work is done in collaboration Dr. Inan's lab

- **Real-Time Vital-Sign Monitoring in the Physical Domain** This work presents a mixed-signal physical-computation-electronics for monitoring three vital signs; namely heart rate, blood pressure, and blood oxygen saturation; from electrocardiography, blood pressure, and photoplethysmography signals in real time. This work is done in collaboration with Hakan Toreyin [138].

- **Learning for VMM+ WTA Embedded Classifiers** This work focuses on learning and hardware implementation using Vector-Matrix Multiplier (VMM) + Winner-Take-All (WTA). The work describes in detail the architecture of the system, learning algorithm and the circuit techniques used to mitigate mismatch in analog system [130, 131].

REFERENCES

[1] K. Kahng and S. Sze, "A floating gate and its application to memory devices," *Electron Devices, IEEE Transactions on*, vol. 14, no. 9, pp. 629–629, 1967.

[2] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (etann) with 10240 'floating gate' synapses," in *International 1989 Joint Conference on Neural Networks*, 1989, pp. 191–196 vol.2.

[3] T. Shibata and T. Ohmi, "A functional mos transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron devices*, vol. 39, no. 6, pp. 1444–1455, 1992.

[4] R. Harrison, J. Bragg, P. Hasler, B. Minch, and S. DeWeerth, "A cmos programmable analog memory-cell array using floating-gate circuits," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 1, pp. 4–11, 2001.

[5] M. Kucic, A. C. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate fourier processor," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 1, pp. 90–99, 2001.

[6] V. Srinivasan, G. Serrano, J. Gray, and P. Hasler, "A precision cmos amplifier using floating-gates for offset cancellation," in *Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005*, 2005, pp. 739–742.

[7] A. Bandyopadhyay, G. J. Serrano, and P. Hasler, "Adaptive algorithm using hot-electron injection for programming analog computational memory elements within 0.2Solid-State Circuits, vol. 41, no. 9, pp. 2107–2114, Sept 2006.

[8] P. Hasler, "Foundation of learning in analog vlsi," Ph.D. dissertation, California Institute of Technology, 1997.

[9] D. W. Graham, E. Farquhar, B. Degnan, C. Gordon, and P. Hasler, "Indirect programming of floating-gate transistors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 5, pp. 951–963, May 2007.

[10] S. Kim, J. Hasler, and S. George, "Integrated floating-gate programming environment for system-level ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–9, 2016.

[11] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 9, pp. 1781–1794, Sept 2010.

[12] C. R. Schlottmann and P. E. Hasler, "A highly dense, low power, programmable analog vector-matrix multiplier: The fpaa implementation," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 403–411, Sept 2011.

[13] R. B. Wunderlich, F. Adil, and P. Hasler, "Floating gate-based field programmable mixed-signal array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 8, pp. 1496–1505, Aug 2013.

[14] B. Rumberg, D. W. Graham, V. Kulathumani, and R. Fernandez, "Hibernets: Energy-efficient sensor networks using analog signal processing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 3, pp. 321–334, Sept 2011.

[15] N. Guo, Y. Huang, T. Mai, S. Patil, C. Cao, M. Seok, S. Sethumadhavan, and Y. Tsividis, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. PP, no. 99, pp. 1–11, 2016.

[16] A. Basu, C. M. Twigg, S. Brink, P. Hasler, C. Petre, S. Ramakrishnan, S. Koziol, and C. Schlottmann, "Rasp 2.8: A new generation of floating-gate based field programmable analog array," in *2008 IEEE Custom Integrated Circuits Conference*, Sept 2008, pp. 213–216.

[17] J. Hasler and H. B. Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers in Neuroscience*, vol. 7, no. 118, 2013.

[18] T. S. Hall, P. Hasler, and D. V. Anderson, *Field-Programmable Analog Arrays: A Floating—Gate Approach*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 424–433. [Online]. Available: https://doi.org/10.1007/3-540-46117-5_45

[19] T. S. Hall, C. M. Twigg, J. D. Gray, P. Hasler, and D. V. Anderson, "Large-scale field-programmable analog arrays for analog signal processing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 11, pp. 2298–2307, Nov 2005.

[20] C. M. Twigg and P. Hasler, "A large-scale reconfigurable analog signal processor (rasp) ic," in *IEEE Custom Integrated Circuits Conference 2006*, Sept 2006, pp. 5–8.

[21] C. Twigg and P. Hasler, "Configurable analog signal processing," *Digital Signal Processing*, vol. 19, no. 6, pp. 904 – 922, 2009, dASP'06 - Defense Applications of Signal Processing. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S105120040700142X

[22] A. Basu, S. Brink, C. Schlottmann, S. Ramakrishnan, C. Petre, S. Koziol, F. Baskaya, C. M. Twigg, and P. Hasler, "A floating-gate-based field-programmable analog array," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 9, pp. 1781–1794, Sept 2010.

[23] C. R. Schlottmann, D. Abramson, and P. E. Hasler, "A mite-based translinear fpaa," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 1–9, Jan 2012.

[24] C. R. Schlottmann, S. Shapero, S. Nease, and P. Hasler, "A digitally enhanced dynamically reconfigurable analog platform for low-power signal processing," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 9, pp. 2174–2184, Sept 2012.

[25] S. George, S. Kim, S. Shah, J. Hasler, M. Collins, F. Adil, R. Wunderlich, S. Nease, and S. Ramakrishnan, "A programmable and configurable mixed-mode FPAA SoC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2253–2261, June 2016.

[26] E. K. F. Lee and P. G. Gulak, "A cmos field-programmable analog array," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 1860–1867, Dec 1991.

[27] R. Wunderlich, "Floating-gate-programmable and reconfigurable, digital and mixed-signal systems," Ph.D. dissertation, Georgia Institute of Technology, 2014.

[28] F. Adil, "Applications of floating-gate based programmable mixed-signal reconfigurable systems," Ph.D. dissertation, Georgia Institute of Technology, 2014.

[29] M. Collins, J. Hasler, and S. George, "An open-source tool set enabling analog-digital-software co-design," *Journal of Low Power Electronics and Applications*, vol. 6, no. 1, p. 3, 2016.

[30] S. Kim, S. Shah, R. Wunderlich, S. George, and J. Hasler, "CAD synthesis tools for large-scale floating-gate fpaa system," *Design Automation for Embedded Systems*, 2017.

[31] R. F. Lyon and C. Mead, "An analog electronic cochlea," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1119–1134, Jul 1988.

[32] L. Watts, D. A. Kerns, R. F. Lyon, and C. A. Mead, "Improved implementation of the silicon cochlea," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 5, pp. 692–700, May 1992.

[33] T. J. Hamilton, C. Jin, A. van Schaik, and J. Tapson, "An active 2-D silicon cochlea," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 2, no. 1, pp. 30–43, March 2008.

[34] S. C. Liu, A. van Schaik, B. A. Minch, and T. Delbruck, "Asynchronous binaural spatial audition sensor with 2 x 64 x 4 channel outputf," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 4, pp. 453–464, Aug 2014.

[35] M. Kucic, A. Low, P. Hasler, and J. Neff, "A programmable continuous-time floating-gate fourier processor," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 90–99, Jan 2001.

[36] T. Delbruck, T. Koch, R. Berner, and H. Hermansky, "Fully integrated 500uW speech detection wake-up circuit," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 2015–2018.

[37] S. Ramakrishnan, A. Basu, L. K. Chiu, J. Hasler, D. Anderson, and S. Brink, "Speech processing on a reconfigurable analog platform," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 430–433, Feb 2014.

[38] Y. Tsividis, "Continuous-time filters in telecommunications chips," *IEEE Communications Magazine*, vol. 39, no. 4, pp. 132–137, Apr 2001.

[39] D. W. Graham, P. E. Hasler, R. Chawla, and P. D. Smith, "A low-power programmable bandpass filter section for higher order filter applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 6, pp. 1165–1176, June 2007.

[40] J. Luu, J. Goeders, M. Wainberg, A. Somerville, T. Yu, K. Nasartschuk, M. Nasr, S. Wang, T. Liu, N. Ahmed, K. B. Kent, J. Anderson, J. Rose, and V. Betz, "VTR 7.0: Next generation architecture and CAD system for FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 7, no. 2, pp. 6:1–6:30, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2617593

[41] P. R. Gray, S. L. P. Hurst, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*. Wiley, 2001.

[42] S. Kim, S. Shah, and J. Hasler, "Calibration of floating-gate soc fpaa system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–9, 2017.

[43] C. Mead, *Analog VLSI and Neural Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[44] E. Rodriguez-Villegas, A. Yufera, and A. Rueda, "A 1.25-V micropower gm-c filter based on fgmos transistors operating in weak inversion," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 100–111, Jan 2004.

[45] A. Veeravalli, E. Sanchez-Sinencio, and J. Silva-Martinez, "A cmos transconductance amplifier architecture with wide tuning range for very low frequency applications," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 6, pp. 776–781, Jun 2002.

[46] B. Rumberg and D. W. Graham, "A low-power and high-precision programmable analog filter bank," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 4, pp. 234–238, April 2012.

[47] O. Omeni, E. Rodriguez-Villegas, and C. Toumazou, "A micropower cmos continuous-time filter with on-chip automatic tuning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 4, pp. 695–705, April 2005.

[48] B. K. Ahuja, H. Vu, C. A. Laber, and W. H. Owen, "A very high precision 500-na cmos floating-gate analog voltage reference," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2364–2372, Dec 2005.

[49] V. Srinivasan, G. J. Serrano, J. Gray, and P. Hasler, "A precision cmos amplifier using floating-gate transistors for offset cancellation," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 2, pp. 280–291, Feb 2007.

[50] R. R. Harrison, J. A. Bragg, P. Hasler, B. A. Minch, and S. P. Deweerth, "A cmos programmable analog memory-cell array using floating-gate circuits," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 1, pp. 4–11, Jan 2001.

[51] S. Moradi and G. Indiveri, "An event-based neural network architecture with an asynchronous programmable synaptic memory," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 1, pp. 98–107, Feb 2014.

[52] C. D. Salthouse and R. Sarpeshkar, "A practical micropower programmable bandpass filter for use in bionic ears," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 1, pp. 63–70, Jan 2003.

[53] S. Hersek, H. Toreyin, and O. T. Inan, "A robust system for longitudinal knee joint edema and blood flow assessment based on vector bioimpedance measurements," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 3, pp. 545–555, June 2016.

[54] H. Toreyin, S. Shah, S. Hersek, O. T. Inan, and J. Hasler, "Proof-of-concept energy-efficient and real-time hemodynamic feature extraction from bioimpedance signals using a mixed-signal field programmable analog array," *International Conference on Biomedical and Health Informatics (BHI)*, 2017.

[55] S. Shah, H. Toreyin, O. T. Inan., and J. Hasler, "Reconfigurable analog classifier for knee-joint rehabilitation," *IEEE Engineering in Medicine and Biology Society*, 2016.

[56] S. Shah and J. Hasler, "Low power speech detector on a fpaa," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[57] C. Teague, S. Hersek, H. Toreyin, M. L. Millard-Stafford, M. L. Jones, G. F. Kogler, M. N. Sawka, and O. T. Inan, "Novel approaches to measure acoustic emissions as biomarkers for joint health assessment," in *Wearable and Implantable Body Sensor Networks (BSN), 2015 IEEE 12th International Conference on*, June 2015, pp. 1–6.

[58] M. Etemadi, O. T. Inan, J. A. Heller, S. Hersek, L. Klein, and S. Roy, "A wearable patch to enable long-term monitoring of environmental, activity and hemodynamics variables," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 2, pp. 280–288, April 2016.

[59] S. Shah, J. Smith, J. Stowell, and J. B. Christen, "Biosensing platform on a flexible substrate," *Sensors and Actuators B: Chemical*, vol. 210, pp. 197 – 203, 2015.

[60] A. Natarajan and J. Hasler, "Modeling, simulation and implementation of circuit elements in an open-source tool set on the fpaa," *Analog Integrated Circuits and Signal Processing*, pp. 1–12, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10470-016-0914-y

[61] C. C. Enz, F. Krummenacher, and E. A. Vittoz, "An analytical mos transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog Integrated Circuits and Signal Processing*, vol. 8, no. 1, pp. 83–114, 1995.

[62] R. Pierret, *Semiconductor Device Fundamentals*. Addison-Wesley, 1996. [Online]. Available: https://books.google.com/books?id=GMZFHwAACAAJ

[63] B. Minch, "Ekv mos transistor model summary." [Online]. Available: http://madvlsi.olin.edu/circuits/handouts

[64] B. P. Degnan, "Temperature robust programmable subthreshold circuits through a balanced force approach," Ph.D. dissertation, Georgia Institute of Technology, 2013.

[65] S. Shah, C. N. Teague, O. T. Inan, and J. Hasler, "A proof-of-concept classifier for acoustic signals from the knee joint on a fpaa," *SENSORS, 2016 IEEE*, 2016.

[66] E. Vittoz and J. Fellrath, "Cmos analog integrated circuits based on weak inversion operations," *IEEE Journal of Solid-State Circuits*, vol. 12, no. 3, pp. 224–231, Jun 1977.

[67] V. Srinivasan, G. Serrano, C. M. Twigg, and P. Hasler, "A floating-gate-based programmable cmos reference," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 11, pp. 3448–3456, Dec 2008.

[68] J. Georgiou and C. Toumazou, "A resistorless low current reference circuit for implantable devices," in *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)*, vol. 3, 2002, pp. III–193–III–196 vol.3.

[69] M. Seok, G. Kim, D. Blaauw, and D. Sylvester, "A portable 2-transistor picowatt temperature-compensated voltage reference operating at 0.5 v," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 10, pp. 2534–2545, Oct 2012.

[70] S. Ramakrishnan and J. Hasler, "Vector-matrix multiply and winner-take-all as an analog classifier," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 22, no. 2, pp. 353–361, Feb 2014.

[71] J. Lazzaro, S. Ryckebusch, M. Mahowald, and C. A. Mead, "Winner-take-all networks of o(n) complexity," in *Advances in Neural Information Processing Systems 1*, D. Touretzky, Ed. Morgan-Kaufmann, 1989, pp. 703–711.

[72] D. D. Dorigo and Y. Manoli, "An ota-c signal processing fpaa with 305 mhz gbw and integrated frequency-independent filter tuning," in *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov 2016, pp. 61–64.

[73] J. Becker, J. Anders, and M. Ortmanns, "A continuous-time field programmable analog array with 1 ghz gbw," in *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Dec 2016, pp. 209–212.

[74] A. Dilello, S. Andryzcik, B. M. Kelly, B. Rumberg, and D. W. Graham, "Temperature compensation of floating-gate transistors in field-programmable analog arrays," *IEEE Internation Symposium on Circuits and System*, 2017.

[75] S. Y. Peng, L. H. Liu, P. K. Chang, T. Y. Wang, and H. Y. Li, "A power-efficient reconfigurable output-capacitor-less low-drop-out regulator for low-power analog sensing front-end," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 6, pp. 1318–1327, June 2017.

[76] V. Higuera, "How is heart disease diagnosed?" *Internet*, 2016. [Online]. Available: https://www.healthline.com/health/heart-disease/tests-diagnosis#modal-close

[77] C. for Disease Control and Prevention, "Heart failure fact sheet," *Internet*, 2016. [Online]. Available: https://www.cdc.gov/dhdsp/data_statistics/fact_sheets/fs_heart_failure.html

[78] A. Garg, D. Virmani, S. Agrawal, C. Agarwal, A. Sharma, G. Stefanini, and J. B. Kostis, "Clinical Application of Biomarkers in Heart Failure with a Preserved Ejection Fraction: A Review," *Cardiology*, vol. 136, no. 3, pp. 192–203, 2017.

[79] A. R. Kemper, W. T. Mahle, G. R. Martin, W. C. Cooley, P. Kumar, W. R. Morrow, K. Kelm, G. D. Pearson, J. Glidewell, S. D. Grosse, and R. R. Howell, "Strategies for implementing screening for critical congenital heart disease," *Pediatrics*, vol. 128, no. 5, pp. e1259–1267, Nov 2011.

[80] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database," *IEEE Trans Biomed Eng*, vol. 33, no. 12, pp. 1157–1165, Dec 1986.

[81] R. Mukkamala, J. O. Hahn, O. T. Inan, L. K. Mestha, C. S. Kim, H. Toreyin, and S. Kyal, "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Theory and Practice," *IEEE Trans Biomed Eng*, vol. 62, no. 8, pp. 1879–1901, Aug 2015.

[82] J. Allen, "Photoplethysmography and its application in clinical physiological measurement," *Physiol Meas*, vol. 28, no. 3, pp. 1–39, Mar 2007.

[83] J. A. Walsh, E. J. Topol, and S. R. Steinhubl, "Novel wireless devices for cardiac monitoring," *Circulation*, vol. 130, no. 7, pp. 573–581, Aug 2014.

[84] Y. T. Zhang and C. C. Poon, "Health informatics: unobtrusive physiological measurement technologies," *IEEE J Biomed Health Inform*, vol. 17, no. 5, p. 893, Sep 2013.

[85] H. J. Baek, G. S. Chung, K. K. Kim, and K. S. Park, "A smart health monitoring chair for nonintrusive measurement of biological signals," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 1, pp. 150–158, Jan 2012.

[86] H. H. Asada, P. Shaltis, A. Reisner, S. Rhee, and R. C. Hutchinson, "Mobile monitoring with wearable photoplethysmographic biosensors," *IEEE Eng Med Biol Mag*, vol. 22, no. 3, pp. 28–40, 2003.

[87] P. Celka, C. Verjus, R. Vetter, P. Renevey, and V. Neuman, "Motion resistant earphone located infrared based heart rate measurement device," *Biomedical Engineering*, 2004.

[88] W.-D. Wang, Z.-B. Zhang, Y.-H. Shen, B.-Q. Wang, and J.-W. Zheng, "Design and implementation of sensing shirt for ambulatory cardiopulmonary monitoring," *Jounal of Medical and Biological Engineering*, 2011.

[89] A. Lanata, E. P. Scilingo, and D. De Rossi, "A multimodal transducer for cardiopulmonary activity monitoring in emergency," *IEEE Trans Inf Technol Biomed*, vol. 14, no. 3, pp. 817–825, May 2010.

[90] L. Turicchia, B. D. Valle, J. L. Bohorquez, W. R. Sanchez, V. Misra, L. Fay, M. Tavakoli, and R. Sarpeshkar, "Ultralow-power electronics for cardiac monitoring," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 9, pp. 2279–2290, Sept 2010.

[91] J. Hasler, "Opportunities in physical computing driven by analog realization," in *2016 IEEE International Conference on Rebooting Computing (ICRC)*, Oct 2016, pp. 1–8.

[92] H. Toreyin, S. Hersek, C. Teague, and O. Inan, "A proof-of-concept system to analyze joint sounds in real time for knee health assessment in uncontrolled settings," *IEEE Sensors Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[93] K. ar Hrlak, Z. F. Erylmaz, M. K. Korkmaz, and H. Treyin, "A proof of-concept wearable photoplethysmography sensor-node for near real-time pulse transit time measurements," *Biomedical Circuits and Systems Conference*, 2017.

[94] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–220, Jun 2000.

[95] A. E. Johnson, T. J. Pollard, L. Shen, L. W. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "MIMIC-III, a freely accessible critical care database," *Sci Data*, vol. 3, p. 160035, May 2016.

[96] B. A. Minch, "Translinear analog signal processing: a modular approach to large-scale analog computation with multiple-input translinear elements," in *Proceedings 20th Anniversary Conference on Advanced Research in VLSI*, Mar 1999, pp. 186–199.

[97] B. A. Minch, C. Diorio, P. Hasler, and C. A. Mead, "Translinear circuits using subthreshold floating-gate mos transistors," *Analog Integrated Circuits and Signal Processing*, vol. 9, no. 2, pp. 167–179, Mar 1996. [Online]. Available: https://doi.org/10.1007/BF00166412

[98] J. W. Severinghaus and Y. Honda, "Pulse oximetry," *Int Anesthesiol Clin*, vol. 25, no. 4, pp. 205–214, 1987.

[99] J. L. Fleg and H. L. Kennedy, "Cardiac arrhythmias in a healthy elderly population: Detection by 24-hour ambulatory electrocardiography," *Chest*, vol. 81, no. 3, pp. 302 – 307, 1982. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0012369215338265

[100] T. Y. Abay and P. A. Kyriacou, "Reflectance Photoplethysmography as Noninvasive Monitoring of Tissue Blood Perfusion," *IEEE Trans Biomed Eng*, vol. 62, no. 9, pp. 2187–2195, Sep 2015.

[101] O. T. Inan, L. Giovangrandi, and G. T. Kovacs, "Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features," *IEEE Trans Biomed Eng*, vol. 53, no. 12 Pt 1, pp. 2507–2515, Dec 2006.

[102] P. Charlton, D. A. Birrenkott, T. Bonnici, M. A. F. Pimentel, A. E. W. Johnson, J. Alastruey, L. Tarassenko, P. J. Watkinson, R. Beale, and D. A. Clifton, "Breathing rate estimation from the electrocardiogram and photoplethysmogram: A review," *IEEE Reviews in Biomedical Engineering*, vol. PP, no. 99, pp. 1–1, 2017.

[103] A. Zbrzeski, P. Hasler, F. Klbl, E. Syed, N. Lewis, and S. Renaud, "A programmable bioamplifier on fpaa for in vivo neural recording," in *2010 Biomedical Circuits and Systems Conference (BioCAS)*, Nov 2010, pp. 114–117.

[104] S. Gabriel, R. W. Lau, and C. Gabriel, "The dielectric properties of biological tissues: II. Measurements in the frequency range 10 Hz to 20 GHz," *Phys Med Biol*, vol. 41, no. 11, pp. 2251–2269, Nov 1996.

[105] J. NYBOER, "Electrical impedance plethysmography; a physical and physiologic approach to peripheral vascular study," *Circulation*, vol. 2, no. 6, pp. 811–821, Dec 1950.

[106] G. Cotter, Y. Moshkovitz, E. Kaluski, A. J. Cohen, H. Miller, D. Goor, and Z. Vered, "Accurate, noninvasive continuous monitoring of cardiac output by whole-body electrical bioimpedance," *Chest*, vol. 125, no. 4, pp. 1431–1440, Apr 2004.

[107] Y. L. Zheng, X. R. Ding, C. C. Poon, B. P. Lo, H. Zhang, X. L. Zhou, G. Z. Yang, N. Zhao, and Y. T. Zhang, "Unobtrusive sensing and wearable devices for health informatics," *IEEE Trans Biomed Eng*, vol. 61, no. 5, pp. 1538–1554, May 2014.

[108] W. G. Kubicek, R. P. Patterson, and D. A. Witsoe, "Impedance cardiography as a noninvasive method of monitoring cardiac function and other parameters of the cardiovascular system," *Ann. New York Acad. Sci*, 1970.

[109] A. Sherwood, M. T. Allen, J. Fahrenberg, R. M. Kelsey, W. R. Lovallo, and L. J. van Doornen, "Methodological guidelines for impedance cardiography," *Psychophysiology*, vol. 27, no. 1, pp. 1–23, Jan 1990.

[110] I. Kononenko, "Machine learning for medical diagnosis: history, state of the art and perspective," *Artificial Intelligence in Medicine*, vol. 23, no. 1, pp. 89 – 109, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S093336570100077X

[111] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct 1990.

[112] S. Y. Peng, M. S. Qureshi, P. E. Hasler, A. Basu, and F. L. Degertekin, "A charge-based low-power high-snr capacitive sensing interface circuit," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 55, no. 7, pp. 1863–1872, Aug 2008.

[113] J. H. Cho, D. R. Morgan, and J. Benesty, "An objective technique for evaluating doubletalk detectors in acoustic echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 6, pp. 718–724, Nov 1999.

[114] G. Cauwenberghs, A. Andreou, J. West, M. Stanacevic, A. Celik, P. Julian, T. Teixeira, C. Diehl, and L. Riddle, "A miniature low-power intelligent sensor node for persistent acoustic surveillance," pp. 294–305, 2005.

[115] T. Delbruck, T. Koch, R. Berner, and H. Hermansky, "Fully integrated 500uw speech detection wake-up circuit," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 2015–2018.

[116] R. Olsson, R. Bogoslovov, and C. Gordon, "Event driven persistent sensing: Overcoming the energy and lifetime limitations in unattended wireless sensors," *IEEE Sensors*, 2016.

[117] P. C. M. Wong, A. K. Uppunda, T. B. Parrish, and S. Dhar, "Cortical mechanisms of speech perception in noise," *Journal of Speech, Language, and Hearing Research*, vol. 51, no. 4, pp. 1026–1041, 2008.

[118] Y. Wu, S. Krishnan, and R. M. Rangayyan, "Computer-aided diagnosis of knee-joint disorders via vibroarthrographic signal analysis: a review," *Crit. Rev. Biomed. Eng.*, vol. 38, pp. 201–24, 2010.

[119] R. Mollan, G. McCullagh, and R. Wilson, "A critical appraisal of auscultation of human joints," *Clinical orthopaedics and related research*, vol. 170, pp. 231–237, 1982.

[120] Y. Zhang, R. Rangayyan, C. Frank, G. Bell, K. Ladly, and Z. Liu, "Classification of knee sound signals using neural networks: a preliminary study," *Proceedings of IASTED Int Symp Expert Systems, Honolulu*, pp. 60–2, 1990.

[121] S. Cai, S. Yang, F. Zheng, M. Lu, Y. Wu, and S. Krishnan, "Knee joint vibration signal analysis with matching pursuit decomposition and dynamic weighted classifier fusion," *Computational and mathematical methods in medicine*, 2013.

[122] H. Toreyin, H. K. Jeong, S. Hersek, C. N. Teague, and O. T. Inan, "Quantifying the consistency of wearable knee acoustical emission measurements during complex motions," *IEEE Journal of Biomedical and Health Informatics*, In press.

[123] C. N. Teague, S. Hersek, H. Toreyin, M. L. Millard-Stafford, M. L. Jones, G. F. Kogler, M. N. Sawka, and O. T. Inan., "Novel methods for sensing acoustical emissions from the knee for wearable joint health assessment," *Transactions on Biomedical Engineering*, 2016.

[124] W. Maass, "On the computational power of winner-take-all," *Neural Computation*, vol. 12, pp. 2519–2535, 2000.

[125] M. Minsky, S. A. Papert, and L. Bottou, *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[126] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59–69, Jan 1982. [Online]. Available: https://doi.org/10.1007/BF00337288

[127] R. Gray, "Vector quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, April 1984.

[128] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *Journal of Machine Learning Research*, 2001.

[129] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE transactions on Audio, Speech, and Language Processing*, 2012.

[130] J. Hasler and S. Shah, "Vmm+ wta embedded classifiers learning algorithm implementable on soc fpaa devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[131] S. Shah and J. Hasler, "Soc fpaa hardware implementation of a vmm + wta embedded learning classifier," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[132] *Analog Devices Low powe differential 12-bit ADC*. [Online]. Available: http://www.analog.com/media/en/technical-documentation/data-sheets/AD7457.pdf

[133] S. Shah and J. Hasler, "Tuning of multiple parameters with a bist system," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. PP, no. 99, pp. 1–9, 2017.

[134] J. Hasler, S. Shah, S. kim, I. Lal, and M. Collins, "Remote FPAA system setup enablingwide accessibility of configurable devices," *Journal of Low Power Electronics and Applications*, vol. Accepted, 2016.

[135] J. Hasler, S. Kim, S. Shah, F. Adil, M. Collins, S. Koziol, and S. Nease, "Transforming mixed-signal circuits class through soc fpaa ic, pcb, and toolset," in *2016 11th European Workshop on Microelectronics Education (EWME)*, May 2016, pp. 1–6.

[136] S. Shah, H. Toreyin, J. Hasler, and A. Natarajan, "Models and techniques for temperature robust systems on a reconfigurable platform," *Journal of Low Power Electronics and Applications*, 2017.

[137] S. S. Shah, H. Toreyin, J. Hasler, and A. Natarajan, "Temperature sensitivity and compensation on a reconfigurable platform," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. PP, no. 99, pp. 1–4, 2017.

[138] S. Shah, H. Toreyin, and J. Hasler, "Real-time vital-sign monitoring in the physical domain on a mixed-signal reconfigurable platform," *Transactions on Biomedical Circuits and System (Submitted)*, 2018.

VITA

Sahil Shah was born in Ahmedabad, India. He received his B.Tech in Electronics and Communication from Manipal Institute of Technology in 2011 and his M.S in Electrical Engineering from Arizona State University in 2014. He received his Ph.D. degree in Electrical and Computer Engineering from Georgia Institute of Technology in 2018. His research interests include low-power circuits and system for real-time processing, wearable devices and biomedical devices.